Knowledge Embedding Based Graph Convolutional Network

Donghan Yu
Carnegie Mellon University
dyu2@cs.cmu.edu

Yiming Yang
Carnegie Mellon University
yiming@cs.cmu.edu

Ruohong Zhang
Carnegie Mellon University
ruohongz@cs.cmu.edu

Yuexin Wu
Google
crickwu@google.com

ABSTRACT

Recently, a considerable literature has grown up around the theme of Graph Convolutional Network (GCN). How to effectively leverage the rich structural information in complex graphs, such as knowledge graphs with heterogeneous types of entities and relations, is a primary open challenge in the field. Most GCN methods are either restricted to graphs with a homogeneous type of edges (e.g., citation links only), or focusing on representation learning for nodes only instead of jointly propagating and updating the embeddings of both nodes and edges for target-driven objectives. This paper addresses these limitations by proposing a novel framework, namely the Knowledge Embedding based Graph Convolutional Network (KE-GCN), which combines the power of GCNs in graph-based belief propagation and the strengths of advanced knowledge embedding (a.k.a. knowledge graph embedding) methods, and goes beyond. Our theoretical analysis shows that KE-GCN offers an elegant unification of several well-known GCN methods as specific cases, with a new perspective of graph convolution. Experimental results on benchmark datasets show the advantageous performance of KE-GCN over strong baseline methods in the tasks of knowledge graph alignment and entity classification[1].

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; *Reasoning about belief and knowledge.*

KEYWORDS

graph convolutional network, knowledge graph, knowledge embedding

[1]Our code is publicly available in https://github.com/PlusRoss/KE-GCN

1 INTRODUCTION

Graph Convolution Networks (GCNs) have received increasing attention in recent machine learning research due as powerful methods for graph-based node feature induction and belief propagation, and been successfully applied to many real-world problems, including natural language processing [14], computer vision [34], recommender systems [43], epidemiological forecasting [38], and more. Existing GCNs share the same core idea, i.e., using a graph to identify the neighborhood of each node, and to learn the embedding (vector representation) of that node via recursive aggregation of the neighborhood embeddings. In other words, graph convolution plays the central role in smoothing the learned representations (latent vectors) of nodes based on belief propagation over the entire graph. However, early work in GCNs also have a limitation in common, i.e., graph convolution is only used to learn node embedding conditioned on fixed edges [12], instead of jointly learning the optimal embeddings for both nodes and edges. Later efforts move on the direction of learning of wights of edges in the input graph in parallel with the learning of node embedding [31, 44], which is more powerful than early GCNs and more capable in model adaptation for downstream tasks. However, those GCNs have a constraint in common, that is, the edges in the input graph must be of homogeneous kind, such as the links in a citation graph or the elements in a co-occurrence count matrix. This constraint (or assumption) significantly limit the applicability of GCNs to a broad range of real-world applications where the capability to model *heterogeneous* relations (edges) is crucial for the true utility of graph-based embedding and inference. Given that heterogeneous types of relations carry rich semantic information in KBs, missing the ability of edge embedding fundamentally limits the expressiveness and prediction power of most GCNs.

As an indirectly related area, methods for Knowledge Graph (KG) completion (a.k.a. knowledge graph embedding) have been intensively studied in recent years. Various algorithms have been developed for joint optimization of the embeddings of both entities and relations, with respect to the task of predicting unknown entity-relation-entity triplets based on observed triplets. Representative approaches include TransE [3], DistMult [41], ComplEx [28], RotatE [24], QuatE [47] and more. The major difference of KG completion methods in contrast to GCNs is that the former do not explicitly leverage the belief propagation power of graph convolution in the representation learning process; instead, entity-relation-entity triplets are treated independently in their objective functions. In other words, those methods lack the ability of using the graph

structures to enforce the local/global smoothness in the embedding spaces for entities and relations.

How to jointly leverage the strengths of both GCN models and KG completion methods for task-oriented representation learning of both entities and relations is an open challenge for research, which has not been studied in sufficient depth and is the focus of this paper. Representative works in this direction, or the only methods of this kind so far to our knowledge, are VR-GCN [42], TransGCN [5], and CompGCN [30]. They use a graph neural network to jointly learn multi-layer latent representations (embeddings) for both entities and relations. Specifically, the entity embedding process recursively aggregates both the neighborhood entity representations and relational representations, which makes sense; However, the *relation* embedding part of the learning process leaves the *entity* representations out of the picture, which is arguably sub-optimal and a fundamental limitation of those models.

To address the aforementioned open challenge and the limitation of existing approaches, we propose a novel framework, namely KE-GCN (Knowledge Embedding based Graph Convolution Network). It provides a theoretically sound generalization of existing GCN models, to allow the incorporation of various knowledge embedding methods for task-oriented embeddings of both entities and relations via graph convolution operations. Especially, in order to capture the rich semantics of heterogeneous relations in knowledge graphs, both entity embeddings and relation embeddings in our model are used to enforce optimization of each other in a recursive aggregation process. Figure 1 illustrates the main differences between previous works and our model. A more detailed comparison will be provided in Section 4. The contributions of our work can be summarized as follows:

- We propose a novel framework KE-GCN which updates both entity and relation embeddings by graph convolution operation leveraging various knowledge embedding techniques.
- KE-GCN originates from a new intuition of GCN, and provides a unified view of several representative methods as its special and restricted cases.
- Experiment results on benchmark datasets for knowledge graph alignment and entity classification tasks show that KE-GCN consistently and significantly outperforms other representative baseline methods.

We organize the rest of this paper as follows: Section 2 introduces related background on graph convolutional networks and knowledge graph embedding methods. Section 3 describes our proposed framework and provides motivation of our method. Section 4 shows how our framework subsumes several representative approaches in a principled way. Section 5 reports our experimental results, followed by the conclusion in Section 6.

2 RELATED BACKGROUND

2.1 Graph Convolutional Network

Graph Convolutional Networks (GCNs) derive its idea from traditional Convolutional Neural Networks (CNNs) by extending convolutional operations onto non-Euclidean data structures. Early trials focus on spectral transformation over adjacency matrices [4, 8] which requires tremendous computational cost of eigen decomposition. Recent work [12] saves the computation of decomposition
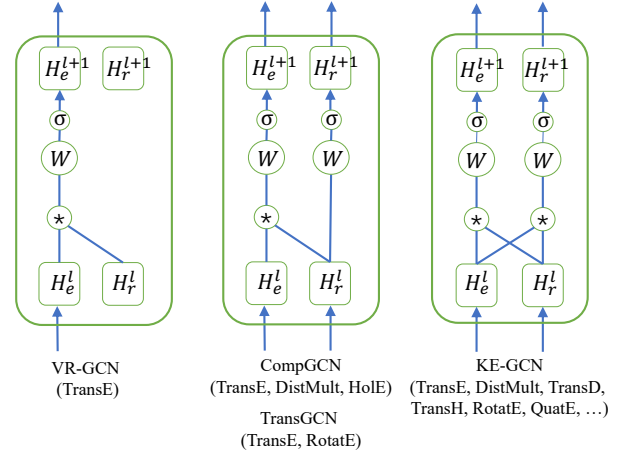
Figure 1: A simple realization of KE-GCN compared to previous works VR-GCN, TransGCN, and CompGCN. H_e^l and H_r^l **means the entity (node) embedding and relation (edge) embedding at layer** l **respectively.** $\star$ **denotes the graph convolution operation which aggregates the neighbour information.** W **is the model parameter of linear transformation and** σ **is the activation function. The names in the brackets below correspond to the incorporated KG completion models.**

by making first order Chebyshev polynomial approximation and popularizes its use while keeping the strong performance over graph-structured data. This change formulates a uniform Message Passing framework for most followup works.

Thereafter, while many extension works tries to improve the effectiveness by reweighting edge weights [31] or making residual links [13], recent works try to leverage the power of GCNs on knowledge graph by allowing multi-relation edge types to interact in the overall framework [16, 21, 22, 30, 33, 46, 48]. These methods define new message passing routines by updating node representation using relation-specific transformation or interaction with relation embeddings. For example, KBAT [16] extends graph attention mechanisms to capture both entity and relation features in a multi-hop neighborhood of a given entity. RGHAT [48] is equipped with relation-level attention and entity-level attention to calculate the weights for neighboring relations and entities respectively. CompGCN [30] leverages the entity-relation composition operations from KE methods like TransE [3] to update entity embeddings. However, none of them have a symmetrical updating rule for their relation embedding, thus limiting the representation power of relation embeddings. Our method differs in that both entity embeddings and relation embeddings are utilized to update each other, where relation embeddings are also enhanced by neighbour context.

2.2 Knowledge Embedding

Traditional Knowledge Graph (KG) tasks focus on making link or entity predictions. These tasks mostly reduce to modeling (head entity, relation, tail entity) triplets. Many solutions [3, 17, 32, 41] rely on learning embeddings for each part. Specifically, a scoring

function f is defined to measure the plausibility of triplets given the embeddings and help update the representation on the training data composed of positive and sampled negative triplets. By using different types of scoring functions, knowledge embedding methods can reflect different designing criteria, which include translation relation [3, 36], inner product [41], rotational relation [24, 47] and many others [9, 23]. For example, TransE [3] assumes that given a triplet, the element-wise addition of head entity and relation embeddings should be close to the tail entity embedding. RotatE [24] is based on the assumption that relation serves as a rotation operation in the complex plane, and it claims to be the first model which can model and infer the following relational patterns: symmetry/anti-symmetry, inversion, and composition.

In our work, we borrow the idea of scoring function by adaptively allowing similar embedding learning design in the new GCN framework. We evaluate the performance of different scoring functions proposed in TransE [3], DistMult [41], TransH [36], TransD [10], RotatE [24], and QuatE [47] through our framework on tasks including knowledge graph alignment and entity classification, which proves the effectiveness of our model design and such incorporation. A more detailed introduction about these methods will be provided in Section 5.1.

3 PROPOSED METHOD

3.1 Reformulation of Vanilla GCN

In vanilla GCN [12], the multi-layer node embedding is updated as follows (we omit the normalization coefficient part for brevity):

$$\mathbf{m}_v^{l+1} = \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^l \qquad (1)$$

$$\mathbf{h}_v^{l+1} = \sigma(W^l(\mathbf{m}_v^{l+1} + \mathbf{h}_v^l)) \qquad (2)$$

where we denote by $\mathbf{h}_v^l$ the embedding of node v at layer l, by $\mathcal{N}(v)$ the set of immediate neighbours of node v, by $\mathbf{m}_v^{l+1}$ the aggregated representation of those neighbors, by $\sigma(\cdot)$ an activation function (e,g., element-wise sigmoid or ReLU), and by W^l the matrix of model parameters to be learned by the GCN. We can reformulate the vanilla GCN by introducing a scoring function f that measures the plausibility of each edge. Edges observed in the graph tend to have higher scores than those that have not been observed. Specifically, for edge (u, v), if we define f as the inner product of the embeddings of the two connected nodes as $f(\mathbf{h}_u, \mathbf{h}_v) = \mathbf{h}_u^T \mathbf{h}_v$, then Equation 1 have the equivalent forms of:

$$\mathbf{m}_v^{l+1} = \sum_{u \in \mathcal{N}(v)} \frac{\partial f(\mathbf{h}_u^l, \mathbf{h}_v^l)}{\partial \mathbf{h}_v^l} = \frac{\partial(\sum_{u \in \mathcal{N}(v)} f(\mathbf{h}_u^l, \mathbf{h}_v^l))}{\partial \mathbf{h}_v^l} \qquad (3)$$

It follows that $\mathbf{h}_v^l + \mathbf{m}_v^{l+1}$ can be regarded as one step gradient ascent to maximize the sum of scoring function $\sum_{u \in \mathcal{N}(v)} f(\mathbf{h}_u^l, \mathbf{h}_v^l)$, with learning rate of 1. Furthermore, Equation 2 can be viewed as a generalized projection onto the embedding space for downstream tasks.

The above reformulation provides an explicit view about *what* the vanilla GCN is optimizing, instead of *how* the updates are executed procedurally. More importantly, it sheds light on how to design a more powerful framework to enable more generalized

multi-relational graph convolution over knowledge graphs, which we introduce in the next section.

3.2 The New Framework

Since the relations in knowledge graphs are of heterogeneous types, we need to define the scoring function f to measure the plausibility of entity-relation-entity triplets, instead of entity-entity pairs in vanilla GCN. Samely, triplets observed in the knowlede graph tend to have higher scores than those that have not been observed. For each triplet (u, r, v), where u, r, v denote head entity, relation, and tail entity respectively, the scoring value is calculated by $f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) \rightarrow \mathbb{R}$ using their embedding vectors. Note that most knowledge embedding techniques can be used to define f.

Analogous to Equation 3, if we denote $\mathbf{h}_v^l$ the embedding of entity v at layer l, the entity updating rules are:

$$\mathbf{m}_v^{l+1} = \sum_{(u,r) \in \mathcal{N}_{\text{in}}(v)} W_r^l \frac{\partial f_{\text{in}}(\mathbf{h}_u^l, \mathbf{h}_r^l, \mathbf{h}_v^l)}{\partial \mathbf{h}_v^l} \qquad (4)$$

$$+ \sum_{(u,r) \in \mathcal{N}_{\text{out}}(v)} W_r^l \frac{\partial f_{\text{out}}(\mathbf{h}_v^l, \mathbf{h}_r^l, \mathbf{h}_u^l)}{\partial \mathbf{h}_v^l} \qquad (5)$$

$$\mathbf{h}_v^{l+1} = \sigma_{\text{ent}}(\mathbf{m}_v^{l+1} + W_0^l \mathbf{h}_v^l) \qquad (6)$$

where $\mathcal{N}_{\text{in}}(v) = \{(u,r) \mid u \xrightarrow{r} v\}$ is the set of immediate entity-relation neighbours of entity v with an in-link while $\mathcal{N}_{\text{out}}(v) = \{(u,r) \mid u \xleftarrow{r} v\}$ is the set of immediate neighbours with an out-link from v. $\mathbf{h}_r^l$ means the embedding of relation r at layer l. Notice that the linear transformation matrix W_r^l is relation-specific, and the scoring functions f_{in} are for the in-link neighbours and f_{out} are for the out-link neighbours, respectively. $\sigma_{\text{ent}}(\cdot)$ denotes the activation function for entity update.

The relation updating rules can be defined in a similar manner:

$$\mathbf{m}_r^{l+1} = \sum_{(u,v) \in \mathcal{N}(r)} \frac{\partial f_r(\mathbf{h}_u^l, \mathbf{h}_r^l, \mathbf{h}_v^l)}{\partial \mathbf{h}_r^l} \qquad (7)$$

$$\mathbf{h}_r^{l+1} = \sigma_{\text{rel}}(W_{\text{rel}}^l(\mathbf{m}_r^{l+1} + \mathbf{h}_r^l)) \qquad (8)$$

where $\mathcal{N}(r) = \{(u,v) \mid u \xrightarrow{r} v\}$ means the set of immediate entity neighbours of relation r, where the left side of tuple is head entity and the right side is tail entity. $\sigma_{\text{rel}}(\cdot)$ denotes the activation function for relation update. Figure 2 provides a demonstration of our model. Note that our framework can subsume other representative methods, which will be introduced in the next section.

Moreover, the derivative $\partial f / \partial \mathbf{h}$ in Equation 5 and Equation 7 can be calculated by ***Auto Differentiation* (AD)** package of many existing libraries including Pytorch [18] and Tensorflow [1], which makes our model easy to implement. Notice that this AD happens in the forward propagation process, which is different from the back-propagation during the training process.

To apply our model on the downstream tasks, denoting the number of layers as L, we use the output entity embedding $\{\mathbf{h}_v^{(L)}\}$ and relation embedding $\{\mathbf{h}_r^{(L)}\}$ of the final layer to construct loss functions. For example, in entity classification task, we use cross-entropy loss based on entity labels; in knowledge graph alignment, we use the distance between the embedding vectors of two entities from different knowledge graphs for loss function. For details of the
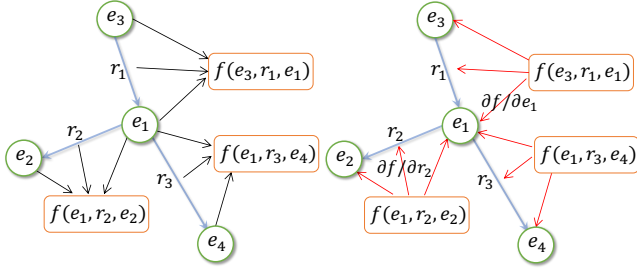
Figure 2: A demonstration for the message passing process of KE-GCN, where e_1, e_2, e_3, e_4 are entities and r_1, r_2, r_3 are relations in a knowledge graph. In one layer of our GCN model, it first calculate the plausibility score of each triplet by certain scoring function f, as shown in the left part, then pass the partial gradient (e.g. $\partial f / \partial e_1$) back to the corresponding entities and relations (e.g. e_1) to update their embeddings, as shown in the right part.

loss functions, please refer to Section 5.2.1 and 5.3.1. The training manners are end-to-end in both tasks. In our experiments for these tasks, we set $f_{in} = f_{out} = f_r$ for simplicity and $W_r^l = W^l$ to prevent over-parameterization. Moreover, to avoid numerical instabilities and exploding/vanishing gradients when used in a deep neural network model, KE-GCN applies the following normalization: we replace $\mathbf{m}_v^{l+1}$ in Equation 6 with $\alpha \mathbf{m}_v^{l+1}/(|\mathcal{N}_{in}(v)| + |\mathcal{N}_{out}(v)|)$ and $\mathbf{m}_r^{l+1}$ in Equation 8 with $\alpha \mathbf{m}_v^{l+1}/|\mathcal{N}(r)|$, where $|\cdot|$ means the cardinality of a set and α is a hyperparameter which controls the weight of aggregated neighbour information.

4 UNIFIED VIEW OF REPRESENTATIVE METHODS

In the following we provide a unified view of several representative GCN methods for knowledge graph modeling, by showing that they are restricted versions under our framework.

4.1 CompGCN

CompGCN [30] is most relevant to our method. In the $(l + 1)$-th layer of CompGCN, the embeddings of each entity and relation are updated as:

- Entity update:

$$\mathbf{m}_v^{l+1} = \sum_{(u,r)\in\mathcal{N}_{in}(v)} W_r^l \phi_{in}\left(\mathbf{h}_u^l, \mathbf{h}_r^l\right)$$
$$+ \sum_{(u,r)\in\mathcal{N}_{out}(v)} W_r^l \phi_{out}\left(\mathbf{h}_u^l, \mathbf{h}_r^l\right) \quad (9)$$
$$\mathbf{h}_v^{l+1} = \sigma(\mathbf{m}_v^{l+1} + W_0^l \mathbf{h}_v^l) \quad (10)$$

where $\phi_{in}, \phi_{out} : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \rightarrow \mathbb{R}^{d_l}$ are the composition operators which can be element-wise subtraction, element-wise multiplication or circular-correlation [19].

- Relation update:

$$\mathbf{h}_r^{l+1} = W_{rel}^l \mathbf{h}_r^l \quad (11)$$

PROPOSITION 1. *CompGCN can be fully recoverd by KE-GCN when 1)* $f_{in}(\mathbf{h}_u^l, \mathbf{h}_r^l, \mathbf{h}_v^l) = \phi_{in}\left(\mathbf{h}_u^l, \mathbf{h}_r^l\right)^T \mathbf{h}_v^l$, $f_{out}(\mathbf{h}_v^l, \mathbf{h}_r^l, \mathbf{h}_u^l) = \phi_{out}\left(\mathbf{h}_u^l, \mathbf{h}_r^l\right)^T \mathbf{h}_v^l$; *and 2)* $f_r = 0$; *and 3)* $\sigma_{rel}(\cdot)$ *is the identity function.*

As shown above, in CompGCN, the relation embedding is only updated by linear transformation. While in KE-GCN, the relation representation update process aggregates neighbour entity representations, shown in Equation 7 and 8, to capture the rich semantics of heterogeneous relations and learn better context-based relation embeddings. Additionally, our framework is more general since the scoring functions $f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v)$ are not restricted to be $\phi(\mathbf{h}_u, \mathbf{h}_r)^T \mathbf{h}_v$, and other forms of knowledge graph embedding techniques such as TransH [36], TransD [10], MLP [9], and NTN [23] can also be incorporated. Note that TransGCN [5] is very similar to CompGCN, and is also subsumed by our model.

4.2 R-GCN

R-GCN [21] extends vanilla GCN with relation-specific linear transformations, without considering relation representations. The embedding update can be listed as follows:

- Entity update:

$$\mathbf{m}_v^{l+1} = \sum_{(u,r)\in\mathcal{N}_{in}(v)} W_r^l \mathbf{h}_u^l + \sum_{(u,r)\in\mathcal{N}_{out}(v)} W_r^l \mathbf{h}_u^l \quad (12)$$
$$\mathbf{h}_v^{l+1} = \sigma(\mathbf{m}_v^{l+1} + W_0^l \mathbf{h}_v^l) \quad (13)$$

- No relation update.

PROPOSITION 2. *R-GCN can be fully recoverd by KE-GCN when 1)* $f_{in}(\mathbf{h}_u^l, \mathbf{h}_r^l, \mathbf{h}_v^l) = f_{out}(\mathbf{h}_v^l, \mathbf{h}_r^l, \mathbf{h}_u^l) = (\mathbf{h}_u^l)^T \mathbf{h}_v^l$; *and 2)* $\mathbf{h}_r^l = 0$ *(no relation embedding).*

4.3 W-GCN

W-GCN [22] treats the relation as learnable weights of edges, and applies vanilla GCN on the weighted simple graph. The update process can be written as:

- Entity update:

$$\mathbf{m}_v^{l+1} = \sum_{(u,r)\in\mathcal{N}_{in}(v)} W^l(\alpha_r^l \mathbf{h}_u^l) + \sum_{(u,r)\in\mathcal{N}_{out}(v)} W^l(\alpha_r^l \mathbf{h}_u^l) \quad (14)$$
$$\mathbf{h}_v^{l+1} = \sigma(\mathbf{m}_v^{l+1} + W^l \mathbf{h}_v^l) \quad (15)$$

where $\alpha_r^l \in \mathbb{R}$ is a relation-specific learnable parameter.
- No relation update.

PROPOSITION 3. *W-GCN can be fully recoverd by KE-GCN when 1)* $f_{in}(\mathbf{h}_u^l, \mathbf{h}_r^l, \mathbf{h}_v^l) = f_{out}(\mathbf{h}_v^l, \mathbf{h}_r^l, \mathbf{h}_u^l) = (\mathbf{h}_u^l)^T \mathbf{h}_v^l$; *and 2)* $W_r^l = W^l \alpha_r^l$; *and 3)* $\mathbf{h}_r^l = 0$ *(no relation embedding).*

5 EXPERIMENTS

5.1 Basic Settings

In this section, we conduct extensive experiments on two well-known tasks of knowledge graphs, *graph alignment* and *entity classification*, to demonstrate the effectiveness of our model. The computing infrastructure we use is NVIDIA RTX 2080Ti GPU in all the experiments. All the experiments are conducted by 5 independent runs with different random seeds. In this paper, following

previous works [21, 30], the input features of entities and relations are random vectors initialized by truncated normal distribution so that our model will rely solely on graph structure. We leave the combination of other features such as text description of entities and relations as future work. Our model is evaluated by combining the following representative knowledge graph embedding methods, with embedding of head entity, relation, and tail entity denoted as $\mathbf{h}_u$, $\mathbf{h}_r$, and $\mathbf{h}_v$ respectively. For each method, we show the corresponding scoring function in KE-GCN as follows:

- TransE [3]: For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{R}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}_u + \mathbf{h}_r - \mathbf{h}_v\|_2^2. \quad (16)$$

- DistMult [41]: For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{R}^d$.

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = \mathbf{h}_u^T \operatorname{diag}(\mathbf{h}_r)\mathbf{h}_v. \quad (17)$$

- TransH [36]: For $\mathbf{h}_u, \mathbf{h}_v \in \mathbb{R}^d, \mathbf{h}_r \in \mathbb{R}^{2d}$, and $\mathbf{h}_{r1}, \mathbf{h}_{r2} \in \mathbb{R}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}'_u + \mathbf{h}_{r2} - \mathbf{h}'_v\|_2^2, \quad (18)$$

$$\mathbf{h}'_u = \mathbf{h}_u - \mathbf{h}_{r1}^T \mathbf{h}_u \mathbf{h}_{r1}, \quad (19)$$

$$\mathbf{h}'_v = \mathbf{h}_v - \mathbf{h}_{r1}^T \mathbf{h}_v \mathbf{h}_{r1}, \quad (20)$$

$$\mathbf{h}_r = [\mathbf{h}_{r1}; \mathbf{h}_{r2}], \quad (21)$$

where $[\cdot; \cdot]$ means concatenation of two vectors.

- TransD [10]: For $\mathbf{h}_u, \mathbf{h}_v, \mathbf{h}_r \in \mathbb{R}^{2d}$, and $\mathbf{h}_{u1}, \mathbf{h}_{u2}, \mathbf{h}_{v1}, \mathbf{h}_{v2}, \mathbf{h}_{r1}, \mathbf{h}_{r2} \in \mathbb{R}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}'_u + \mathbf{h}_{r2} - \mathbf{h}'_v\|_2^2, \quad (22)$$

$$\mathbf{h}'_u = \mathbf{h}_{u1} + \mathbf{h}_{u2}^T \mathbf{h}_{u1} \mathbf{h}_{r1}, \quad (23)$$

$$\mathbf{h}'_v = \mathbf{h}_{v1} - \mathbf{h}_{v2}^T \mathbf{h}_{v1} \mathbf{h}_{r1}, \quad (24)$$

$$\mathbf{h}_u = [\mathbf{h}_{u1}; \mathbf{h}_{u2}], \mathbf{h}_v = [\mathbf{h}_{v1}; \mathbf{h}_{v2}], \mathbf{h}_r = [\mathbf{h}_{r1}; \mathbf{h}_{r2}], \quad (25)$$

where $[\cdot; \cdot]$ means concatenation of two vectors.

- RotatE [24]: For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{C}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}_u \circ \mathbf{h}_r - \mathbf{h}_v\|_2^2, \quad (26)$$

where $\circ$ denotes element-wise product and the modulus of any element in $\mathbf{h}_r$ is 1, i.e. $|\mathbf{h}_r[i]| = 1 \ \forall i \in \{1, 2, \cdots, d\}$. The norm of complex vector is defined as $\|\mathbf{v}\|_p = \sqrt[p]{\sum |\mathbf{v}_i|^p}$.

- QuatE [47]: For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{H}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = \mathbf{h}_u \otimes \mathbf{h}_r \cdot \mathbf{h}_v, \quad (27)$$

where $\otimes$ and $\cdot$ denote Hamilton Product and Inner Product in the hypercomplex space respectively.

Note that TransE and DistMult are widely used knowledge graph embedding techniques since they are simple and effective. RotatE and QuatE are recent proposed models and achieves state-of-the-art results in commonly used benchmark datasets for knowledge graph completion. TransH and TransD are the extensions of TransE, and can not be incorporated by previous works. We leave other embedding methods for future work.

5.2 Knowledge Graph Alignment

Knowledge graph alignment refer to the task which aims to find entities in two different knowledge graphs KG_1 and KG_2 that represent the same real-world entity. To apply our GCN framework on knowledge graph alignment, we follow the previous work [35] which utilize two GCNs with shared parameters to model two

Table 1: Dataset statistics of DBP15K for knowledge graph alignment task, including number of entities, relations, triplets of each knowledge graph.

Datasets		#Entities	#Relations	#Triplets
DBP$_{\text{ZH-EN}}$	Chinese	66,469	2,830	153,929
	English	98,125	2,317	237,674
DBP$_{\text{JA-EN}}$	Japanese	65,744	2,043	164,373
	English	95,680	2,096	233,319
DBP$_{\text{FR-EN}}$	French	66,858	1,379	192,191
	English	105,889	2,209	278,590

knowledege graphs separately, then the output embeddings $\mathbf{h}$ of each final layer are used for entity alignment. To align from KG_1 to KG_2, for a specific entity/relation u in KG_1, we compute the L1-distance between u and each entity/relation v in KG_2 using their embeddings $\mathbf{h}_u$ and $\mathbf{h}_v$, and returns a ranked list of entities/relations as candidate alignments based on the distances. The alignment can be also performed from KG_2 to KG_1. In the experiments, we report the averaged results of both directions of KG alignment.

5.2.1 Loss Function. We denote the training entity alignment set as $S = \{(u, v)\}$, where u is the entity in KG_1 while entity v belongs to KG_2 and they refer to the same real-world entity. For the loss function, we follow previous work [35] to use margin-based ranking loss:

$$\mathcal{L} = \sum_{(u,v) \in S} \sum_{(u',v') \in S'_{(u,v)}} l(u, v, u', v') \quad (28)$$

$$l(u, v, u', v') = [\|\mathbf{h}_u - \mathbf{h}_v\|_1 + \gamma - \|\mathbf{h}_{u'} - \mathbf{h}_{v'}\|_1]_+ \quad (29)$$

where $[x]_+ = \max\{0, x\}$. $S'_{(u,v)}$ denotes the set of negative entity alignments constructed by corrupting (u, v), i.e. replacing u or v with a randomly chosen entity in KG_1 or KG_2. γ means the margin hyper-parameter separating positive and negative entity alignments. Following [35], we randomly choose 5 negtive alignments for each positive alignment and set $\gamma = 3$.

5.2.2 Dataset. We use DBP15K [25] which contains three datasets built from multi-lingual DBpedia, namely DBP$_{\text{ZH-EN}}$ (Chinese - English), DBP$_{\text{ZH-EN}}$ (Japanese - English) and DBP$_{\text{FR-EN}}$ (French - English) for knowledge graph alignment. A summary statistics of these datasets is provided in Table 1. Each dataset has 15, 000 reference entity alignment. Following previous work [6, 27, 35], we randomly split 30% of them for training and the rest for testing. We further set aside 30% of the training set as a validation set for hyperparameter tuning, and retrain the model on the whole training set to obtain test performance.

5.2.3 Baselines. To demonstrate the effectiveness of KE-GCN, we compare it with several representative multi-relational GCN baseline methods including R-GCN [21], W-GCN [22], KBGAT [16] and CompGCN [30]. We also include other baseline methods which are designed specifically for knowledge graph alignment task for a comprehensive comparison: MTransE [7], IPTransE [49], JAPE [25], AlignE [26], GCN-Align [35], MuGNN [6], and AliNet [27]. Since our model solely relies on structure information, we do not compare

with the alignment models which incorporate the surface information of entities into their representations [37, 40]. Besides, some heterogeneous-graph-based models [33, 45, 46] which can not scale to knowledge graphs with hundreds or thousands of edge types are ruled out in comparison.

5.2.4 Implementation Detail. After performing grid search for hyperparameters, we set the learning rate to be 0.01, the hidden dimension to be 200, the number of layers to be 4, and $\alpha = 0.3$. The activation function is set as ReLU. We train our model in full-batch setting using Adam [11]. Following previous work [6, 27], we report Hits@1, Hits@10, Mean Reciprocal Rank (MRR) to evaluate the entity alignment performance.

5.2.5 Results of Entity Alignment. Table 2 shows the experiment results in DBP15K datasets comparing with all the baseline methods. Our model achieves the best or highly competitive results in all the three datasets, outperforming the baseline methods by a large margin. Specifically, our model outperforms the best baseline methods by 5.7%, 6.5%, and 6.6% in MRR on DBP$_{\text{ZH-EN}}$, DBP$_{\text{JA-EN}}$, and DBP$_{\text{FR-EN}}$ respectively. Note that even equipped with the recent state-of-the-art knowledge graph embedding methods, i.e., RotatE [24] and QuatE [47], CompGCN [30] still obtains much lower performance than KE-GCN. A more clear comparison is shown in Table 3, where we present the results on DBP$_{\text{ZH-EN}}$ dataset of CompGCN and KE-GCN incorporated with TransE and QuatE. The results demonstrate the effectiveness of graph convolution updating for relation embeddings.

We also report the performance of our proposed model with different knowledge graph embedding techniques including TransE, DistMult, TransH, TransD, RotatE, and QuatE on DBP$_{\text{ZH-EN}}$, DBP$_{\text{JA-EN}}$, and DBP$_{\text{FR-EN}}$ datasets as shown in Table 4, Table 5, and Table 6 respectively. Note that the choice of embedding techniques does have a large impact on the performance, and QuatE achieves the best results, which is reasonable since it also outperforms other methods in knowledge graph completion task, and satisfies the essential of relational representation learning (i.e., modeling symmetry, anti-symmetry and inversion relation).

5.2.6 Results of Relation Alignment. We also test on relation alignment task to demonstrate the importance of our proposed relation embedding update process. Since the number of reference aligned relations is very small, we train the model on the entity alignment task mentioned above and directly use the trained relation embedding for relation alignment (as zero-shot evaluation). The results over 5 different runs are shown in Table 7, where we see that our model significantly outperforms CompGCN [30] and VR-GCN [42]. This demonstrate the importance of incorporating entity representation into the update of relation embeddings. We do not compare with R-GCN [21] and W-GCN [22] since relation embeddings are not involved in their models.

5.3 Knowledge Graph Entity Classification

Entity Classification is the task of predicting the labels of entities in a given knowledge graph. We follow previous work [21, 30] to use the entity output of the last layer in KE-GCN for label classification.

5.3.1 Loss Function. In this paper we conduct experiments on both multi-class classification and multi-label classification. Denoting N as the number of entities, C as the number of classes, and $\mathcal{Y}_L$ as the set of entity indices that have labels. For multi-class classification, we use the following loss:

$$\mathcal{L} = - \sum_{u \in \mathcal{Y}_L} \sum_{c=1}^{C} Y_{uc} \ln \widehat{Y}_{uc} \tag{30}$$

where $Y_{uc} = 1$ means that the true class of entity u is c, otherwise $Y_{uc} = 0$. $\widehat{Y} \in \mathbb{R}^{N \times C}$ is the output of GCN model, which comes after a row-wise softmax function. For multi-label classification, we use:

$$\mathcal{L} = - \sum_{u \in \mathcal{Y}_L} \sum_{c=1}^{C} \left[Y_{uc} \ln \widehat{Y}_{uc} + (1 - Y_{uc}) \ln(1 - \widehat{Y}_{uc}) \right] \tag{31}$$

$Y_{uc} = 1$ means that entity u contains label c, otherwise $Y_{uc} = 0$. $\widehat{Y} \in \mathbb{R}^{N \times C}$ is the output of GCN model, which comes after an element-wise sigmoid function.

5.3.2 Dataset. We conduct experiments on the following datasets: AM [20] which contains relationship between different artifacts in Amsterdam Museum, WN [3, 29] which consists of a collection of triplets (synset, relation, synset) extracted from WordNet 3.0 [15], FB15K [3, 39] which is extracted from a typical large-scale knowledge graph Freebase [2]. The statistics of these datasets is shown in Table 8. Each entity in AM and WN datasets has at most one label while entities in FB15K can have multiple labels[2]. For AM dataset, we follow the train/test split convention [20, 21]. As for WN and FB15K datasets, we randomly split the labeled entities into train/valid/test by the ratio of 10%/10%/80%.

5.3.3 Baselines. We compare KE-GCN with vanilla GCN [12] and the relation-based GCN models including R-GCN [21], W-GCN [22], KBGAT [16], and CompGCN [30]. For vanilla GCN, we transform the multi-relational graphs to homogenous graphs, by setting the weight of edge between two entities as the number of their relations.

5.3.4 Implementation Detail. After performing grid search for hyperparameters, we set learning rate to be 0.01, hidden dimension to be 32, number of layers to be 4, and $\alpha = 0.3$. The activation function is set as ReLU. We train our model in full-batch setting using Adam [11]. In AM and WN, Accuracy are reported to evaluate the entity classification performance. While in FB15K, we report Precision@1 (P@1), Precision@5 (P@5) and NDCG@5 (N@5).

5.3.5 Results. The experiment results over 5 different runs on AM and WN datasets are shown in Table 9, where the average of classification accuracy is reported. The results on FB15K dataset under metrics of P@1, P@5, N@5 are shown in Table 10. From these results, KE-GCN outperforms all the baseline GCN methods, which demonstrate the effectiveness of our proposed model in entity classification task, including multi-class classification and multi-label classification. We report the best results of CompGCN [30] and KE-GCN incorporated with different knowledge graph embedding methods, where combining TransE [3] achieves the highest performance. The results of KE-GCN incorporated with different KE methods are shown in Table 11 and Table 12. Additionally, we see

---

[2]Please refer to [29] and [39] for details of label collection in WN and FB15K datasets respectively.

Table 2: Experiment results in knowledge graph entity alignment task on DBP15K datasets, where the average results over 5 different runs are reported. * indicate that results are directly taken from [27]. The results of VR-GCN [42] are directly taken from the original paper. CompGCN marked with † incorporates the composition operations in RotatE [24] and QuatE [47] while original CompGCN [30] only contains subtraction, multiplication and circular-correlation operations.

Models	DBP$_{ZH-EN}$			DBP$_{JA-EN}$			DBP$_{FR-EN}$		
	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
MTransE*[7]	0.364	30.8	61.4	0.349	27.9	57.5	0.335	24.4	55.6
IPTransE*[49]	0.516	40.6	73.5	0.474	36.7	69.3	0.451	33.3	68.5
JAPE*[25]	0.490	41.2	74.5	0.476	36.3	68.5	0.430	32.4	66.7
AlignE*[26]	0.581	47.2	79.2	0.563	44.8	78.9	0.599	48.1	82.4
GCN-Align*[35]	0.549	41.3	74.4	0.546	39.9	74.5	0.532	37.3	74.5
MuGCN*[6]	0.611	49.4	**84.4**	0.621	50.1	**85.7**	0.621	49.5	87.0
AliNet*[27]	0.628	53.9	82.6	0.645	54.9	83.1	0.657	55.2	85.2
R-GCN*[21]	0.564	46.3	73.4	0.571	47.1	75.4	0.570	46.9	75.8
W-GCN [22]	0.553	43.6	73.8	0.554	41.2	74.7	0.541	39.8	74.4
VR-GCN [42]	0.501	38.0	73.3	0.470	35.2	72.2	0.495	36.1	75.1
KBGAT [16]	0.582	48.0	77.3	0.582	47.6	77.7	0.593	47.4	80.9
CompGCN[30]	0.605	49.4	81.2	0.614	50.4	82.2	0.625	50.5	85.0
CompGCN†	0.628	52.8	81.1	0.629	52.8	81.5	0.641	52.6	85.4
KE-GCN	**0.664**	**56.2**	84.2	**0.670**	**57.0**	85.2	**0.683**	**57.2**	**88.5**

Table 3: Direct comparison of CompGCN and KE-GCN incorporated with the same knowledge graph embedding techniques (TransE [3] and QuatE [47]) on DBP$_{ZH-EN}$.

Model	MRR	H@1	H@10
CompGCN (TransE)	0.605 ± 0.003	49.4 ± 0.4	81.2 ± 0.3
KE-GCN (TransE)	**0.648 ± 0.003**	**54.3 ± 0.3**	**83.4 ± 0.3**
CompGCN (QuatE)	0.628 ± 0.003	52.8 ± 0.3	81.1 ± 0.4
KE-GCN (QuatE)	**0.664 ± 0.004**	**56.2 ± 0.4**	**84.2 ± 0.4**

Table 4: Knowledge graph entity alignment results over 5 different runs on DBP$_{ZH-EN}$ by incorporating different knowledge graph embedding methods into our model.

KE-GCN (X)	MRR	H@1	H@10
X = TransE	0.648 ± 0.003	54.3 ± 0.3	83.4 ± 0.3
X = TransH	0.650 ± 0.003	54.3 ± 0.4	**84.4 ± 0.3**
X = DistMult	0.621 ± 0.003	52.0 ± 0.4	80.3 ± 0.4
X = TransD	0.635 ± 0.003	53.1 ± 0.3	82.7 ± 0.4
X = RotatE	0.653 ± 0.004	54.9 ± 0.4	83.8 ± 0.4
X = QuatE	**0.664 ± 0.004**	**56.2 ± 0.4**	84.2 ± 0.4

Table 5: Knowledge graph entity alignment results over 5 different runs on DBP$_{JA-EN}$ dataset by incorporating different knowledge graph embedding methods into our model.

KE-GCN (X)	MRR	H@1	H@10
X = TransE	0.652 ± 0.003	54.8 ± 0.3	84.3 ± 0.3
X = TransH	0.654 ± 0.003	54.6 ± 0.4	**85.5 ± 0.3**
X = DistMult	0.622 ± 0.004	51.7 ± 0.5	81.2 ± 0.3
X = TransD	0.652 ± 0.001	54.5 ± 0.2	85.4 ± 0.2
X = RotatE	0.659 ± 0.002	55.7 ± 0.2	85.0 ± 0.3
X = QuatE	**0.670 ± 0.001**	**57.0 ± 0.2**	85.2 ± 0.3

Table 6: Knowledge graph entity alignment results over 5 different runs on DBP$_{FR-EN}$ dataset by incorporating different knowledge graph embedding methods into our model.

KE-GCN (X)	MRR	H@1	H@10
X = TransE	0.669 ± 0.002	55.9 ± 0.2	87.5 ± 0.2
X = TransH	0.673 ± 0.002	56.1 ± 0.2	87.7 ± 0.2
X = DistMult	0.640 ± 0.002	52.4 ± 0.2	84.7 ± 0.2
X = TransD	0.660 ± 0.002	54.2 ± 0.2	87.6 ± 0.1
X = RotatE	0.673 ± 0.002	56.0 ± 0.3	88.2 ± 0.2
X = QuatE	**0.683 ± 0.002**	**57.2 ± 0.3**	**88.5 ± 0.2**

Table 7: Knowledge graph relation alignment results over 5 different runs on DBP15K datasets. All the models are incorporated with the same KG completion method TransE [3].

Model	DBP$_{ZH-EN}$	DBP$_{JA-EN}$	DBP$_{FR-EN}$
VR-GCN	0.352 ± 0.006	0.335 ± 0.008	0.280 ± 0.017
KBGAT	0.341 ± 0.010	0.330 ± 0.013	0.274 ± 0.022
CompGCN	0.366 ± 0.007	0.347 ± 0.009	0.284 ± 0.015
KE-GCN	**0.514 ± 0.006**	**0.466 ± 0.011**	**0.412 ± 0.021**

Table 8: Number of entities, relations, edges and classes along with the number of labeled entities for each dataset in entity classification task. Labeled denotes the subset of entities that have labels and that are to be classified. Classes denotes the total number of categories of labels.

Datasets	AM	WN	FB15K
#Entities	1,666,764	40,551	14,904
#Relations	133	18	1,341
#Triplets	5,988,321	145,966	579,654
#Labeled	1,000	31,943	13,445
#Classes	11	24	50

Table 9: The mean and standard deviation of classification accuracy over 5 different runs on AM and WN datasets for multi-class classification task. * indicates the results that are directed taken from [30].

Models	AM	WN
GCN	86.2 ± 1.4	53.4 ± 0.2
R-GCN	89.3[*]	55.1 ± 0.6
W-GCN	90.2 ± 0.9[*]	54.2 ± 0.5
KBGAT	85.7 ± 1.7	53.7 ± 1.1
CompGCN	90.6 ± 0.2[*]	55.9 ± 0.4
KE-GCN	**91.2 ± 0.2**	**57.8 ± 0.5**

Table 10: The mean and standard deviation of Precision@1 (P@1), Precision@5 (P@5), NDCG@5 (N@5) over 5 different runs on FB15K dataset for multi-label classification task.

Models	P@1	P@5	N@5
GCN	86.1 ± 0.3	69.0 ± 0.3	82.7 ± 0.2
R-GCN	91.7 ± 0.6	73.0 ± 0.4	89.5 ± 0.6
W-GCN	91.2 ± 0.6	72.8 ± 0.3	88.6 ± 0.5
KBGAT	90.5 ± 0.7	72.4 ± 0.8	87.5 ± 0.8
CompGCN	92.5 ± 0.1	74.0 ± 0.3	90.1 ± 0.2
KE-GCN	**94.3 ± 0.2**	**74.7 ± 0.2**	**91.6 ± 0.2**

that vanilla GCN performs worse than any relational GCNs in all the datasets, which demonstrates that relation modelling is significant for entity classification.

Table 11: Entity classification accuracy results over 5 different runs on AM and WN datasets by incorporating different knowledge graph embedding methods into our model.

KE-GCN (X)	AM	WN
X = TransE	**91.2 ± 0.2**	**57.8 ± 0.5**
X = TransH	90.5 ± 0.3	57.4 ± 0.3
X = DistMult	89.5 ± 0.4	56.4 ± 0.1
X = TransD	90.1 ± 0.2	57.1 ± 0.2
X = RotatE	90.6 ± 0.4	56.6 ± 0.3
X = QuatE	91.0 ± 0.4	56.9 ± 0.3

Table 12: Entity classification results over 5 different runs on FB15K dataset by incorporating different knowledge graph embedding methods into our model.

KE-GCN (X)	P@1	P@5	N@5
X = TransE	94.3 ± 0.2	**74.7 ± 0.2**	**91.6 ± 0.2**
X = TransH	94.3 ± 0.3	74.6 ± 0.2	**91.6 ± 0.2**
X = DistMult	94.2 ± 0.1	72.4 ± 0.4	89.9 ± 0.3
X = TransD	93.9 ± 0.4	74.5 ± 0.2	91.3 ± 0.2
X = RotatE	93.8 ± 0.4	73.7 ± 0.2	90.5 ± 0.2
X = QuatE	**94.9 ± 0.4**	74.2 ± 0.3	**91.6 ± 0.2**

6 CONCLUSION

This paper presents a novel framework which leverages various knowledge graph embedding methods into GCNs for multi-relational graph modelling, and more importantly, update both entity and relation representation using graph convolution operation. We show that our model originates from a new intuition behind graph convolution in the view of generalized projected gradient ascent, and subsumes other representative methods as its special and restricted cases. Experiments show that the proposed model obtains the state-of-the-art results in benchmark datasets of two well-known tasks: knowledge graph alignment and entity classification. In the future, we plan to apply our framework into a broader range of domains containing knowledge graphs, including Q&A, recommender system, computer vision and time series analysis. It's also worth exploring to go beyond triplets and extend our framework for knowledge hypergraphs.

ACKNOWLEDGMENTS

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. 265–283.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 1247–1250.

[3] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 2787–2795. https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6203

[5] Ling Cai, Bo Yan, Gengchen Mai, Krzysztof Janowicz, and Rui Zhu. 2019. TransGCN: Coupling transformation assumptions with graph convolutional networks for link prediction. In *Proceedings of the 10th International Conference on Knowledge Capture*. 131–138.

[6] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. 2019. Multi-Channel Graph Neural Network for Entity Alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1452–1461. https://doi.org/10.18653/v1/P19-1140

[7] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 1511–1517. https://doi.org/10.24963/ijcai.2017/209

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 3837–3845. https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html

[9] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani (Eds.). ACM, 601–610. https://doi.org/10.1145/2623330.2623623

[10] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. 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, Beijing, China, 687–696. https://doi.org/10.3115/v1/P15-1067

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[13] Guohao Li, Matthias Müller, Ali K. Thabet, and Bernard Ghanem. 2019. DeepGCNs: Can GCNs Go As Deep As CNNs?. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 9266–9275. https://doi.org/10.1109/ICCV.2019.00936

[14] Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1506–1515. https://doi.org/10.18653/v1/D17-1159

[15] George A. Miller. 1992. WordNet: A Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*. https://www.aclweb.org/anthology/H92-1116

[16] Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4710–4723. https://doi.org/10.18653/v1/P19-1466

[17] Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 1955–1961. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12484

[18] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[19] Lawrence R Rabiner and Bernard Gold. 1975. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.* (1975).

[20] Petar Ristoski, Gerben Klaas Dirk De Vries, and Heiko Paulheim. 2016. A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In *International Semantic Web Conference*. Springer, 186–194.

[21] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[22] Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 3060–3067. https://doi.org/10.1609/aaai.v33i01.33013060

[23] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 926–934. https://proceedings.neurips.cc/paper/2013/hash/b337e84de8752b27eda3a12363109e80-Abstract.html

[24] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=HkgEQnRqYQ

[25] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*. Springer, 628–644.

[26] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 4396–4402. https://doi.org/10.24963/ijcai.2018/611

[27] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2019. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. *arXiv preprint arXiv:1911.08936* (2019).

[28] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, 2071–2080. http://proceedings.mlr.press/v48/trouillon16.html

[29] Ke Tu, Peng Cui, Xiao Wang, Fei Wang, and Wenwu Zhu. 2018. Structural Deep Embedding for Hyper-Networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 426–433. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16797

[30] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=BylA_C4tPr

[31] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[32] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[33] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 2022–2032. https://doi.org/10.1145/3308558.3313562

[34] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2018. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829* (2018).

[35] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 349–357. https://doi.org/10.18653/v1/D18-1032

[36] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, Carla E. Brodley and Peter Stone (Eds.). AAAI Press, 1112–1119. http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531

[37] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 5278–5284. https://doi.org/10.24963/ijcai.2019/733

[38] Yuexin Wu, Yiming Yang, Hiroshi Nishiura, and Masaya Saitoh. 2018. Deep Learning for Epidemiological Predictions. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 1085–1088. https://doi.org/10.1145/3209978.3210077

[39] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation Learning of Knowledge Graphs with Entity Descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 2659–2665. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12216

[40] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3156–3161. https://doi.org/10.18653/v1/P19-1304

[41] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6575

[42] Rui Ye, Xin Li, Yujie Fang, Hongyu Zang, and Mingzhong Wang. 2019. A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 4135–4141. https://doi.org/10.24963/ijcai.2019/574

[43] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 974–983. https://doi.org/10.1145/3219819.3219890

[44] Donghan Yu, Ruohong Zhang, Zhengbao Jiang, Yuexin Wu, and Yiming Yang. 2019. Graph-Revised Convolutional Network. *arXiv preprint arXiv:1911.07123* (2019).

[45] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2019. Graph Transformer Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 11960–11970. https://proceedings.neurips.cc/paper/2019/hash/9d63484abb477c97640154d40595a3bb-Abstract.html

[46] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 793–803. https://doi.org/10.1145/3292500.3330961

[47] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion Knowledge Graph Embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 2731–2741. https://proceedings.neurips.cc/paper/2019/hash/d961e9f236177d65d21100592edb0769-Abstract.html

[48] Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhi-Ping Shi, Hui Xiong, and Qing He. 2020. Relational Graph Neural Network with Hierarchical Attention for Knowledge Graph Completion.. In *AAAI*. 9612–9619.

[49] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative Entity Alignment via Joint Knowledge Embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 4258–4264. https://doi.org/10.24963/ijcai.2017/595