Generalized Multi-Relational Graph Convolution Network

Donghan Yu [1] **Yiming Yang** [1] **Ruohong Zhang** [1] **Yuexin Wu** [1]

Abstract

Graph Convolutional Networks (GCNs) have received increasing attention in recent machine learning. 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 optimizing the embeddings of both nodes and edges for target-driven objectives. This paper addresses these limitations by proposing a novel framework, namely the GEneralized Multi-relational Graph Convolutional Networks (GEM-GCN), which combines the power of GCNs in graph-based belief propagation and the strengths of advanced knowledge-base embedding methods, and goes beyond. Our theoretical analysis shows that GEM-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 GEM-GCN over strong baseline methods in the tasks of knowledge graph alignment and entity classification.

1. Introduction

Graph Convolution Networks (GCNs) have received increasing attention in recent machine learning research 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 (Kipf & Welling, 2016; Marcheggiani & Titov, 2017), computer vision (Wang et al., 2018a; Landrieu & Simonovsky, 2018), recommender systems (Monti et al., 2017; Ying et al., 2018), epidemiological forecasting (Wu et al., 2018), 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. However, most existing GCN models 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.

As an indirectly related area, methods for Knowledge Graph (KG) completion (a.k.a. knowledge graph embedding) have been intensively studied in recent years. Representative approaches include TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019a), QuatE (Zhang et al., 2019) 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.

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 (Ye et al., 2019) and CompGCN (Vashishth et al., 2019). They use a graph neural network to jointly learn multi-layer latent representations (embeddings) for both entities and relations. However, the *relation embedding* part of the learning process leaves the entity representations out of the picture. In other words, the graph structure is only used to propagate information from embedded nodes plus edges in the optimization of node embedding (which makes sense), but not used to propagate information from embedded nodes in the optimization of edge embedding, which is arguably sub-optimal and a fundamen-

[1]Carnegie Mellon University, Pittsburgh, PA 15213, USA. Correspondence to: Donghan Yu <dyu2@cs.cmu.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

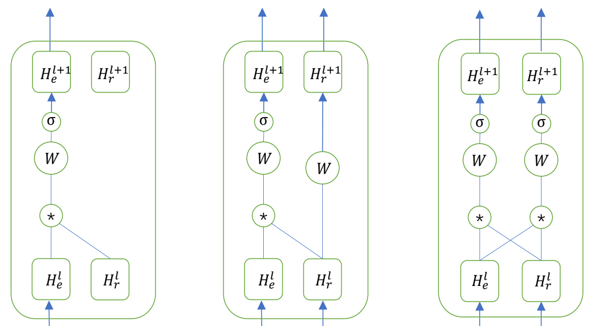

Figure 1. A simple realization of GEM-GCN compared to previous works VR-GCN and COMPGCN, where H_e^l and H_r^l means the entity (node) embedding and relation (edge) embedding at layer l respectively. $\star$ denotes the graph convolution operation which aggregates the neighbour information. W is the model parameter of linear transformation and σ is the activation function. The names in the brackets below correspond to the incorporated KG completion models.

tal limitation of those models. Moreover, the existing works did not leverage recent advanced KG completion methods such as RotatE (Sun et al., 2019a) and QuatE (Zhang et al., 2019).

To address the aforementioned open challenge and the limitations of existing approaches, we propose novel framework, namely GEM-GCN (GEneralized Multi-relational Graph Convolution Network). It provides a theoretically sound generalization of existing GCN models, to allow the incorporation of various KG completion 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. Experimental results on benchmark datasets for knowledge graph alignment and entity classification tasks show that GEM-GCN consistently and significantly outperforms other representative baseline methods.

2. Proposed Method

2.1. Reformulation of Vanilla GCN

In vanilla GCN (Kipf & Welling, 2016), the multi-layer node embedding is updated as follows (we omit the normalization coefficient part for brevity):

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

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

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:

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

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.

2.2. 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 $f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v)$ using their embedding vectors. Note that most knowledge graph 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:

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

$$\mathbf{h}_v^{l+1} = \sigma_{\text{ent}}(\mathbf{m}_v^{l+1} + W_0^l \mathbf{h}_v^l) \tag{5}$$

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:

$$\mathbf{m}_r^{l+1} = \sum_{(u,v)\in\mathcal{N}(r)} \frac{\partial f_r(\mathbf{h}_u^l, \mathbf{h}_r^l, \mathbf{h}_v^l)}{\partial \mathbf{h}_r^l} \tag{6}$$

$$\mathbf{h}_r^{l+1} = \sigma_{\text{rel}}(W_{\text{rel}}^l(\mathbf{m}_r^{l+1} + \mathbf{h}_r^l)) \tag{7}$$

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. Notice that our framework is very general and can subsume other representative methods, which is introduced in Appendix A.

Moreover, the derivative $\partial f/\partial \mathbf{h}$ in Equation 4 and Equation 6 can be calculated by *Auto Differentiation* (AD) package of many existing libraries including Pytorch (Paszke et al., 2017) and Tensorflow (Abadi et al., 2016), which makes our model easy to implement. Notice that this AD happens in the inference process, instead of backpropagation 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 loss functions, please refer to Appendix C.2 and E.3. The training manners are end-to-end in both tasks.

3. Experiments

3.1. 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. Note that due to the page limit, we show the results of entity classification in Appendix E. In this paper, following previous works (Schlichtkrull et al., 2018; Vashishth et al., 2019), 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. Our model is evaluated by combining the following representative knowledge graph embedding methods: TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), TransH (Wang et al., 2014), TransD (Ji et al., 2015) RotatE (Sun et al., 2019a) and QuatE (Zhang et al., 2019), where the details are shown in Appendix B. We leave other embedding methods for future work.

3.2. Knowledge Graph Alignment

Knowledge graph alignment refers to the task which aims to find entities/relations in two different knowledge graphs KG_1 and KG_2 that represent the same real-world entity/relation. A detailed description can be found in Appendix C. Note that we only show the main results of entity alignment in the following sections while leaving relation alignment in Appendix D.

3.2.1. Datasets and Baselines

We use DBP15K (Sun et al., 2017) which contains three datasets built from multi-lingual DBpedia, namely $\text{DBP}_{\text{ZH-EN}}$ (Chinese-English), $\text{DBP}_{\text{ZH-EN}}$ (Japanese-English) and $\text{DBP}_{\text{FR-EN}}$ (French-English) for knowledge graph alignment. A summary statistics of these datasets is provided in Appendix C.1. Following previous work (Cao et al., 2019; Sun et al., 2019b), we report Hits@1, Hits@10, Mean Reciprocal Rank (MRR) to evaluate the alignment performance.

To demonstrate the effectiveness of GEM-GCN, we compare it with several representative multi-relational GCN baseline methods including R-GCN (Schlichtkrull et al., 2018), W-GCN (Shang et al., 2019), VR-GCN (Ye et al., 2019) and CompGCN (Vashishth et al., 2019). We also include other baseline methods which are designed specifically for knowledge graph alignment task for a comprehensive comparison: MTransE (Chen et al., 2017), IPTransE (Zhu et al., 2017), JAPE (Sun et al., 2017), AlignE (Sun et al., 2018), GCN-Align (Wang et al., 2018b), MuGNN (Cao et al., 2019), and AliNet (Sun et al., 2019b). Note that since our model solely relies on structure information, we do not compare with the alignment models which

Table 1. 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 (Sun et al., 2019b). The results of VR-GCN (Ye et al., 2019) are directly taken from the original paper. The highest scores are marked in **bold**. COMPGCN marked with † incorporates the composition operations in RotatE (Sun et al., 2019a) and QuatE (Zhang et al., 2019) while original COMPGCN (Vashishth et al., 2019) 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*(Chen et al., 2017)	0.364	30.8	61.4	0.349	27.9	57.5	0.335	24.4	55.6
IPTransE*(Zhu et al., 2017)	0.516	40.6	73.5	0.474	36.7	69.3	0.451	33.3	68.5
JAPE*(Sun et al., 2017)	0.490	41.2	74.5	0.476	36.3	68.5	0.430	32.4	66.7
AlignE*(Sun et al., 2018)	0.581	47.2	79.2	0.563	44.8	78.9	0.599	48.1	82.4
GCN-Align*(Wang et al., 2018b)	0.549	41.3	74.4	0.546	39.9	74.5	0.532	37.3	74.5
MuGCN*(Cao et al., 2019)	0.611	49.4	**84.4**	0.621	50.1	**85.7**	0.621	49.5	87.0
AliNet*(Sun et al., 2019b)	0.628	53.9	82.6	0.645	54.9	83.1	0.657	55.2	85.2
R-GCN*(Schlichtkrull et al., 2018)	0.564	46.3	73.4	0.571	47.1	75.4	0.570	46.9	75.8
W-GCN (Shang et al., 2019)	0.553	43.6	73.8	0.554	41.2	74.7	0.541	39.8	74.4
VR-GCN (Ye et al., 2019)	0.501	38.0	73.3	0.470	35.2	72.2	0.495	36.1	75.1
COMPGCN(Vashishth et al., 2019)	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
GEM-GCN	**0.664**	**56.2**	84.2	**0.670**	**57.0**	85.2	**0.683**	**57.2**	**88.5**

incorporate the surface information of entities into their representations like (Xu et al., 2019; Wu et al., 2019).

3.2.2. RESULTS OF ENTITY ALIGNMENT

Table 1 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_{ZH-EN}, DBP_{JA-EN}, and DBP_{FR-EN} respectively. Note that even equipped with the recent state-of-the-art knowledge graph embedding methods, i.e., RotatE (Sun et al., 2019a) and QuatE (Zhang et al., 2019), COMPGCN (Vashishth et al., 2019) still obtains much lower performance than GEM-GCN. 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 (Bordes et al., 2013), DistMult (Yang et al., 2014), TransH (Wang et al., 2014), TransD (Ji et al., 2015), RotatE (Sun et al., 2019a), and QuatE (Zhang et al., 2019), as shown in Table 2. Note that the choice of embedding techniques does have a large impact on the performance, and QuatE (Zhang et al., 2019) 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).

Table 2. Knowledge graph entity alignment results over 5 different runs on DBP_{ZH-EN} by incorporating different knowledge graph embedding methods into our model. The highest scores are marked in **bold**.

Model	MRR	H@1	H@10
GEM-GCN (TransE)	0.648 ± 0.003	54.3 ± 0.3	83.4 ± 0.3
GEM-GCN (TransH)	0.650 ± 0.003	54.3 ± 0.4	$\mathbf{84.4 \pm 0.3}$
GEM-GCN (DistMult)	0.621 ± 0.003	52.0 ± 0.4	80.3 ± 0.4
GEM-GCN (TransD)	0.635 ± 0.003	53.1 ± 0.3	82.7 ± 0.4
GEM-GCN (RotatE)	0.653 ± 0.004	54.9 ± 0.4	83.8 ± 0.4
GEM-GCN (QuatE)	$\mathbf{0.664 \pm 0.004}$	$\mathbf{56.2 \pm 0.4}$	84.2 ± 0.4

4. 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.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795, 2013.

Cao, Y., Liu, Z., Li, C., Li, J., and Chua, T.-S. Multi-channel graph neural network for entity alignment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1452–1461, 2019.

Chen, M., Tian, Y., Yang, M., and Zaniolo, C. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1511–1517, 2017.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 601–610, 2014.

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. 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)*, pp. 687–696, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Landrieu, L. and Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567, 2018.

Marcheggiani, D. and Titov, I. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1506–1515, 2017.

Miller, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5115–5124, 2017.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Rabiner, L. R. and Gold, B. Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1975.

Ristoski, P., De Vries, G. K. D., and Paulheim, H. A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In *International Semantic Web Conference*, pp. 186–194. Springer, 2016.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

Shang, C., Tang, Y., Huang, J., Bi, J., He, X., and Zhou, B. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3060–3067, 2019.

Socher, R., Chen, D., Manning, C. D., and Ng, A. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pp. 926–934, 2013.

Sun, Z., Hu, W., and Li, C. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, pp. 628–644. Springer, 2017.

Sun, Z., Hu, W., Zhang, Q., and Qu, Y. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pp. 4396–4402, 2018.

Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019a.

Sun, Z., Wang, C., Hu, W., Chen, M., Dai, J., Zhang, W., and Qu, Y. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. *arXiv preprint arXiv:1911.08936*, 2019b.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pp. 2071–2080, 2016.

Tu, K., Cui, P., Wang, X., Wang, F., and Zhu, W. Structural deep embedding for hyper-networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Vashishth, S., Sanyal, S., Nitin, V., and Talukdar, P. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018a.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI conference on artificial intelligence*, 2014.

Wang, Z., Lv, Q., Lan, X., and Zhang, Y. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 349–357, 2018b.

Wu, Y., Yang, Y., Nishiura, H., and Saitoh, M. Deep learning for epidemiological predictions. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1085–1088. ACM, 2018.

Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., and Zhao, D. Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv preprint arXiv:1908.08210*, 2019.

Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Xu, K., Wang, L., Yu, M., Feng, Y., Song, Y., Wang, Z., and Yu, D. Cross-lingual knowledge graph alignment via graph matching neural network. *arXiv preprint arXiv:1905.11605*, 2019.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Ye, R., Li, X., Fang, Y., Zang, H., and Wang, M. A vectorized relational graph convolutional network for multi-relational network alignment. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4135–4141, 2019.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983. ACM, 2018.

Zhang, S., Tay, Y., Yao, L., and Liu, Q. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, pp. 2731–2741, 2019.

Zhu, H., Xie, R., Liu, Z., and Sun, M. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*, pp. 4258–4264, 2017.

A. Unified View of Representative Methods

In the following we provide a unified view of several representative GCN methods for knowledge graph modeling (Schlichtkrull et al., 2018; Shang et al., 2019; Vashishth et al., 2019), by showing that they are restricted versions under our framework.

A.1. CompGCN

CompGCN (Vashishth et al., 2019) is most relevant to our method. We denote the set of immediate neibourghs of entity v as $\mathcal{N}(v) = \{(u, r)\}$ where u is the entity connected with v by relation r. In the $(l+1)$-th layer of CompGCN, the embeddings of each entity and relation are updated as:

- Entity update:

$$\mathbf{m}_v^{l+1} = \sum_{(u,r)\in\mathcal{N}_{\text{in}}(v)} W_r^l \phi\left(\mathbf{h}_u^l, \mathbf{h}_r^l\right) + \sum_{(u,r)\in\mathcal{N}_{\text{out}}(v)} W_r^l \phi\left(\mathbf{h}_u^l, \mathbf{h}_r^l\right) \tag{8}$$

$$\mathbf{h}_v^{l+1} = \sigma(\mathbf{m}_v^{l+1} + W_0^l \mathbf{h}_v^l) \tag{9}$$

  where $\phi : \mathbb{R}^{d_l} \times \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$ is a composition operator which can be element-wise subtraction, element-wise multiplication or circular-correlation (Rabiner & Gold, 1975).

- Relation update:

$$\mathbf{h}_r^{l+1} = W_{\text{rel}}^l \mathbf{h}_r^l \tag{10}$$

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

As shown above, in CompGCN, the relation embedding is only updated by linear transformation. While in GEM-GCN, the relation representation update process aggregates neighbour entity representations, shown in Equation 6 and 7, 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 $\phi\left(\mathbf{h}_u, \mathbf{h}_r\right)^T \mathbf{h}_v$, and other forms of knowledge graph embedding techniques such as TransH (Wang et al., 2014), TransD (Ji et al., 2015), MLP (Dong et al., 2014), and NTN (Socher et al., 2013) can also be incorporated.

A.2. R-GCN

R-GCN (Schlichtkrull et al., 2018) extends vanilla GCN with relation-specific linear transformations, without considering relation representations. The embedding update can be listed as follows:

- Entity update:

$$\mathbf{m}_v^{l+1} = \sum_{(u,r)\in\mathcal{N}_{\text{in}}(v)} W_r^l \mathbf{h}_u^l + \sum_{(u,r)\in\mathcal{N}_{\text{out}}(v)} W_r^l \mathbf{h}_u^l \tag{11}$$

$$\mathbf{h}_v^{l+1} = \sigma(\mathbf{m}_v^{l+1} + W_0^l \mathbf{h}_v^l) \tag{12}$$

- No relation update.

Proposition 2 *R-GCN can be fully recoverd by GEM-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).*

A.3. W-GCN

W-GCN (Shang et al., 2019) 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:

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

$$\mathbf{h}_v^{l+1} = \sigma(\mathbf{m}_v^{l+1} + W_0^l \mathbf{h}_v^l) \tag{14}$$

  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 GEM-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).*

B. Scoring Functions of Knowledge Graph Embedding Methods

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 GEM-GCN, which may be slightly different with the original scoring function.

- TransE (Bordes et al., 2013): For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{R}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}_u + \mathbf{h}_r - \mathbf{h}_v\|_2^2. \tag{15}$$

- DistMult (Yang et al., 2014): For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{R}^d$.

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = \mathbf{h}_u^T \operatorname{diag}(\mathbf{h}_r) \mathbf{h}_v. \tag{16}$$

- TransH (Wang et al., 2014): For $\mathbf{h}_u, \mathbf{h}_v \in \mathbb{R}^d, \mathbf{h}_r \in \mathbb{R}^{2d}$, and $\mathbf{h}_{r1}, \mathbf{h}_{r2} \in \mathbb{R}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}'_u + \mathbf{h}_{r2} - \mathbf{h}'_v\|_2^2, \quad (17)$$
$$\mathbf{h}'_u = \mathbf{h}_u - \mathbf{h}_{r1}^T \mathbf{h}_u \mathbf{h}_{r1}, \quad (18)$$
$$\mathbf{h}'_v = \mathbf{h}_v - \mathbf{h}_{r1}^T \mathbf{h}_v \mathbf{h}_{r1}, \quad (19)$$
$$\mathbf{h}_r = [\mathbf{h}_{r1}; \mathbf{h}_{r2}], \quad (20)$$

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

- TransD (Ji et al., 2015): For $\mathbf{h}_u, \mathbf{h}_v, \mathbf{h}_r \in \mathbb{R}^{2d}$, and $\mathbf{h}_{u1}, \mathbf{h}_{u2}, \mathbf{h}_{v1}, \mathbf{h}_{v2}, \mathbf{h}_{r1}, \mathbf{h}_{r2} \in \mathbb{R}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}'_u + \mathbf{h}_{r2} - \mathbf{h}'_v\|_2^2, \quad (21)$$
$$\mathbf{h}'_u = \mathbf{h}_{u1} + \mathbf{h}_{u2}^T \mathbf{h}_{u1} \mathbf{h}_{r1}, \quad (22)$$
$$\mathbf{h}'_v = \mathbf{h}_{v1} - \mathbf{h}_{v2}^T \mathbf{h}_{v1} \mathbf{h}_{r1}, \quad (23)$$
$$\mathbf{h}_u = [\mathbf{h}_{u1}; \mathbf{h}_{u2}], \mathbf{h}_v = [\mathbf{h}_{v1}; \mathbf{h}_{v2}], \mathbf{h}_r = [\mathbf{h}_{r1}; \mathbf{h}_{r2}], \quad (24)$$

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

- RotatE (Sun et al., 2019a): For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{C}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = -\|\mathbf{h}_u \circ \mathbf{h}_r - \mathbf{h}_v\|_2^2, \quad (25)$$

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 (Zhang et al., 2019): For $\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v \in \mathbb{H}^d$,

$$f(\mathbf{h}_u, \mathbf{h}_r, \mathbf{h}_v) = \mathbf{h}_u \otimes \mathbf{h}_r \cdot \mathbf{h}_v, \quad (26)$$

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

C. Details of Knowledge Graph Alignment Task

To apply our GCN framework on knowledge graph alignment, we follow the previous work (Wang et al., 2018b) which utilize two GCNs with shared parameters to model two knowledege graphs separately, then the output embeddings of each final layer are used for entity/relation alignment. More specifically, to align from one knowledge graph KG_1 to another knowledge graph 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 output from each final layer $\mathbf{h}_u$ 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. Particularly,

1) For entity alignment task, following previous work (Wang et al., 2018b; Cao et al., 2019; Sun et al., 2019b), we randomly split 30% of aligned entities 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.

2) 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).

C.1. Data Statistics

DBP15K (Sun et al., 2017) contains three datasets built from multi-lingual DBpedia, namely $\text{DBP}_{\text{ZH-EN}}$ (Chinese-English), $\text{DBP}_{\text{ZH-EN}}$ (Japanese-English) and $\text{DBP}_{\text{FR-EN}}$ (French-English) for knowledge graph alignment. A summary statistics of these datasets is shown in Table 3.

Table 3. Dataset statistics of DBP15K for knowledge graph alignment task, including number of entities, relations, triplets of each knowledge graph and the number of aligned entities and relations.

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

Datasets		#Aligned Entities	#Aligned Relations
$\text{DBP}_{\text{ZH-EN}}$	Chinese English	15,000	891
$\text{DBP}_{\text{JA-EN}}$	Japanese English	15,000	582
$\text{DBP}_{\text{FR-EN}}$	French English	15,000	75

C.2. 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 (Wang et al., 2018b) to use margin-based ranking loss:

$$\mathcal{L} = \sum_{(u,v)\in S} \sum_{(u',v')\in S'_{(u,v)}} l(u, v, u', v') \quad (27)$$

$$l(u, v, u', v') = [\|\mathbf{h}_u - \mathbf{h}_v\|_1 + \gamma - \|\mathbf{h}_{u'} - \mathbf{h}_{v'}\|_1]_+ \quad (28)$$

Table 4. Knowledge graph relation alignment results over 5 different runs on DBP15K datasets. All the models are incorporated with the same KG completion method TransE (Bordes et al., 2013).The highest score are marked in **bold**.

Model	DBP_{ZH-EN}	DBP_{JA-EN}	DBP_{FR-EN}
VR-GCN	0.352 ± 0.006	0.335 ± 0.008	0.280 ± 0.017
COMPGCN	0.366 ± 0.007	0.347 ± 0.009	0.284 ± 0.015
GEM-GCN	$\mathbf{0.514 \pm 0.006}$	$\mathbf{0.466 \pm 0.011}$	$\mathbf{0.412 \pm 0.021}$

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. Suppose for each positive entity alignment, we randomly choose k negtive alignments. Following (Wang et al., 2018b), we set $\gamma = 3$ and $k = 5$.

C.3. Implementation Details

We perform grid search for hyperparameters in the following values: the learning rate l_r in $\{0.001, 0.005, 0.01, 0.05\}$, α in $\{0.1, 0.2, \cdots, 0.9\}$, the hidden dimension d for entity and relation in $\{50, 100, 200, 300\}$, the number of layers L in $\{1, 2, 3, 4, 5\}$. The final selected setting is $l_r = 0.01$, $\alpha = 0.3$, $d = 200$, and $L = 4$. The activation function is set as ReLU. We train our model in full-batch setting using Adam (Kingma & Ba, 2014).

D. Results of Relation Alignment

To directly test the effect of our proposed relation embedding update process, we use the relation embedding for the relation alignment task mentioned in Appendix C. The results over 5 different runs are shown in Table 4, where we see that our model significantly outperforms COMPGCN (Vashishth et al., 2019) and VR-GCN (Ye et al., 2019). This demonstrate the importance of incorporating entity representation into the update of relation embeddings. Note that we do not compare with R-GCN (Schlichtkrull et al., 2018) and W-GCN (Shang et al., 2019) since relation embeddings are not involved in their models.

E. Knowledge Graph Entity Classification

Entity Classification is the task of predicting the labels of entities in a given knowledge graph. We follow previous work (Schlichtkrull et al., 2018; Vashishth et al., 2019) to use the entity output of the last layer in GEM-GCN for label classification.

E.1. Datasets and Baselines

We conduct experiments on the following datasets: AM (Ristoski et al., 2016), WN (Bordes et al., 2013; Tu et al., 2018), and FB15K (Bordes et al., 2013; Xie et al., 2016). The details and statistics of these datasets are shown in Appendix E.2. Each entity in AM and WN datasets has at most one label while entities in FB15K can have multiple labels[1]. 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).

We compare GEM-GCN with vanilla GCN (Kipf & Welling, 2016) and the relation-based GCN models including R-GCN (Schlichtkrull et al., 2018), W-GCN (Shang et al., 2019) and COMPGCN (Vashishth et al., 2019). For vanilla GCN, we transform the multi-relational graphs to simple graphs with only entities, by setting the weight of edge between two entities as the number of their relations.

E.2. Data Statistics

AM (Ristoski et al., 2016) contains relationship between different artifacts in Amsterdam Museum. WN (Bordes et al., 2013; Tu et al., 2018) consists of a collection of triplets (synset, relation, synset) extracted from WordNet 3.0 (Miller, 1995). FB15K (Bordes et al., 2013; Xie et al., 2016) is extracted from a typical large-scale knowledge graph Freebase (Bollacker et al., 2008) . The statistics of these datasets are shown in Table 5.

For AM dataset, we follow the train/test split convention as (Ristoski et al., 2016; Schlichtkrull et al., 2018). As for WN and FB15K datasets, we randomly split the labeled entities into train/valid/test by the ratio of 10%/10%/80% for semi-supervised learning.

Table 5. 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

[1]Please refer to (Tu et al., 2018) and (Xie et al., 2016) for details of label collection in WN and FB15K datasets respectively.

E.3. 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:

$$\mathcal{L} = -\sum_{u \in \mathcal{Y}_L} \sum_{c=1}^{C} Y_{uc} \ln \widehat{Y}_{uc} \tag{29}$$

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:

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

$Y_{uc} = 1$ 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.

E.4. Implementation Details

Grid search are conducted for hyperparameters in the following values: the learning rate l_r in $\{0.001, 0.005, 0.01, 0.05\}$, α in $\{0.1, 0.2, \cdots, 0.9\}$, the hidden dimension d for entity and relation in $\{8, 16, 32, 64\}$, the number of layers L in $\{1, 2, 3, 4, 5\}$. The final selected setting is $l_r = 0.01$, $\alpha = 0.3$, $d = 32$, and $L = 4$. The activation function is set as ReLU. We train our model in full-batch setting using Adam (Kingma & Ba, 2014).

E.5. Results

The experiment results over 5 different runs are shown in Table 6, 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 7. From these results, GEM-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 (Vashishth et al., 2019) and GEM-GCN incorporated with different knowledge graph embedding methods, and surprisingly, combining TransE (Bordes et al., 2013) achieves the highest performance. Additionally, we see that vanilla GCN performs worse than any relational GCNs in all the datasets, which demonstrates that relation modelling is significant for entity classification.

Table 6. 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 (Vashishth et al., 2019).

Models	AM	WN
GCN	86.2 ± 1.4	53.4 ± 0.2
R-GCN	89.3^*	55.1 ± 0.6
W-GCN	$90.2 \pm 0.9^*$	54.2 ± 0.5
CompGCN	$90.6 \pm 0.2^*$	55.9 ± 0.4
GEM-GCN	$\mathbf{91.2} \pm \mathbf{0.2}$	$\mathbf{57.8} \pm \mathbf{0.5}$

Table 7. 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
CompGCN	92.5 ± 0.1	74.0 ± 0.3	90.1 ± 0.2
GEM-GCN	$\mathbf{94.3} \pm \mathbf{0.2}$	$\mathbf{74.7} \pm \mathbf{0.2}$	$\mathbf{91.6} \pm \mathbf{0.2}$