# Knowledge Augmented Attention Gating Embedding for Link Prediction

Zewei Chen*, Shuhong Chen*, Chen Li†, Xianwei Zheng†, Minfan He† and Xutao Li*
* Key Lab of Digital Signal and Image Processing of Guangdong Province, Department of Electronic Engineering, Shantou University, Shantou, 515063, China
E-mail: 22zwchen@stu.edu.cn, 18shchen3@stu.edu.cn, lixt@stu.edu.cn
† School of Mathematics, Foshan University, Foshan, 52800, China
E-mail: fslichen@fosu.edu.cn, alexwzheng@fosu.edu.cn, heminfan1980@126.com

***Abstract*—Knowledge graphs (KGs) use graphical representations to illustrate relations between entities, forming a structured knowledge base. However, KGs often suffer from incompleteness, lacking some entities and their interactions. Current link prediction technologies based on KG embeddings primarily focus on learning entity features and relation patterns from existing KGs to address the incompleteness issue. Yet, these methods encounter the inherent sparsity issue in KGs—uneven distribution of entity connections and scarcity of direct links. This complexity challenges models in identifying intricate, potential associations, thereby compromising link prediction accuracy and overall model performance. In this paper, we introduce a innovative Knowledge Augmented Attention-based Gating Embedding (KAAGE) framework for link prediction. Our model augments knowledge by reversing relations and triples, then integrates both original and augmented structural data through gating units. Extensive experiments on three established benchmark datasets demonstrate the superiority of KAAGE over current models. Our implementation code is available at https://github.com/22zwChen/KAAGE**

## I. Introduction

In recent years, KGs have played a critical role in numerous domains and caught the attention of various research communities, including information retrieval [1], [2], question answering systems [3], [4], and recommendation systems [5], [6]. These sophisticated repositories take the form of graphs, encapsulating real-world entities and the complex networks of relations that bind them together. Entities are represented as nodes, symbolizing tangible objects or conceptual abstractions found within reality, while relations serve as the interlinking fabric delineating the connections among these entities within the KG structure.

Despite their extensive coverage of relations, entities, and triples, current KGs still struggle with issues of pronounced data sparsity, diverse relational categories, and intricate hierarchies. These challenges have spurred a wave of advanced research aimed at improving knowledge graph completion. Early logic-based approaches, such as HL-MRFs [7] and CPRA [8], are limited by data sparsity and poor scalability. Distributed representation schemes were then proposed to translate entities and relations into points within a continuous vector space, where mathematical operations can be used to evaluate entity similarities, as demonstrated by RESCAL [9] and TransE [6]. With the remarkable ability to discern features, neural networks have become prominent in the field of Knowledge Graph Embedding (KGE) learning. Neural-network-driven methods, including DSKG [10] and KGQA [11], autonomously learn representations by applying non-linear transformations to remodel the feature distribution of input data from its original space to a distinct feature space. However, despite the significant progress made by these methodologies, most do not strategically employ effective data augmentation techniques to address the incompleteness of data. Additionally, they often handle knowledge facts in isolation, overlooking the structural context inherently woven into the fabric of KGs.

In response to these challenges, we advocate for the integration of Graph Data Augmentation to enrich both the initial and augmented structural information of KGs. Building on the principles of CompGCN [12], we allow the information in the directed edges of the knowledge graph to propagate bidirectionally. We then incorporate an efficient gating mechanism that facilitates the harmonious integration of both the original structural information and the augmented structural information. Our key contributions are summarized as follows:

- To enrich knowledge graph, we implement knowledge augmentation strategies, specifically including the consideration of inverse relations and triples, which are integrated with the knowledge graph to counteract its sparseness.
- A gating unit is incorporated to merge the original structural information with the augmented ones, to dynamically assessing how important each piece of information is.
- Massive experimental results indicate that KAAGE framework outperforms competitors on three standard benchmark datasets, particularly on the Kinship dataset.

The remainders of this paper are structured as follows: Section II delves into the core of the matter, detailing the key aspects and constituent parts. Section III provides an examination of the proposed framework, mapping out the essential methodologies that define its operation. Section IV is dedicated to the empirical substantiation of KAAGE to validate the framework's effectiveness. In Section V, we encapsulate the key contributions of this research and underscore their importance.

## II. Preliminaries

### A. Problem Definition

***Knowledge Graph***. We define $\mathcal{E}$, $\mathcal{R}$ as the sets of entities $e$ and relations $r$, respectively. A knowledge graph is denoted by $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ where $\mathcal{T} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the set of triples. Each triple in $\mathcal{T}$ can be represented as $(h, r, t)$ where $h \in \mathcal{E}$ is head entity, $t \in \mathcal{E}$ is tail entity and $r \in \mathcal{R}$ is relation.

***Link Prediction***. To achieve link prediction, we compute score for each triple $(h, r, t)$ by our model. Specifically, our model predicts the missing one in an incomplete triple such as $(h, r, ?)$ by a pre-designed scoring function $\psi(h, r, t) \in \mathbb{R}$.

### B. ConvE

ConvE, initially introduced by Dettmers et al. [13], revolutionized the approach to link prediction in KGs. This model leverages 2D convolutional operations and stacks of nonlinear transformations to accurately predict missing links within KGs. The scoring function for a triple within this framework can be formally expressed as follows:

$$\Psi(e_i, r_k, e_j) = f(\text{vec}(f([\mathbf{h}_i; \mathbf{g}_k] * \omega))\mathbf{W})e_j \tag{1}$$

where $e_i$, $e_j$, $r_k \in \mathbb{R}^{k_h k_w \times 1}$ represent the embeddings of the head entity, tail entity, and relation, respectively. $\mathbf{h}_i, \mathbf{g}_k \in \mathbb{R}^{k_h \times k_w}$ are 2D reshaped representations of head entity and relation. A 2D convolution $*$ with kernel $\omega$ is applied to $[\mathbf{h}_i; \mathbf{g}_k]$, followed by activation $f$ and linear transformation $\mathbf{W}$. Given its effectiveness, ConvE is adopted as the decoder in our model.

### C. Gating unit

To balance diverse features for KG embedding, we incorporate an adaptive, learnable gating unit that dynamically merges them as follow:

$$\mathbf{g} = \sigma(\mathbf{W_{ge}} \mathbf{e_i} + \mathbf{W_{ga}} \mathbf{a} + \mathbf{b}), \tag{2}$$

$$\mathbf{u} = \tau(\mathbf{W_u}(\mathbf{e_i} \oplus \mathbf{a})), \tag{3}$$

$$\mathbf{e} = \mathbf{g} \odot \mathbf{u} + (\mathbf{1} - \mathbf{g}) \odot \mathbf{e_i}, \tag{4}$$

where $e_i \in \mathbb{R}^{d_1}$, $a \in \mathbb{R}^{d_2}$ are input embeddings. This unit employs linear transformations $\mathbf{W_u} \in \mathbb{R}^{d_1 \times (d_1 + d_2)}$, $\mathbf{W_{ge}} \in \mathbb{R}^{d_1 \times d_1}$, and $\mathbf{W_{ga}} \in \mathbb{R}^{d_1 \times d_2}$, along with a bias $\mathbf{b}$. $\sigma$ is the *sigmoid* function, and $\tau$ is hyperbolic tangent function. Importantly, the gating unit handles inputs of differing dimensions and outputs $e \in \mathbb{R}^{d_1}$, which matches the dimension of $e_i$, defined as $e = gate(e_i, a)$.

## III. Methodology

Figure 1 shows each attention head of our framework KAAGE and multi-head attention mechanism is applied to our experiments on partial datasets.

### A. Knowledge Augmentation

To facilitate knowledge augmentation, we view each triple $t_{ij}^k = (e_i, r_k, e_j) \in \mathcal{T}$ as a fundamental unit of knowledge and enhance it by reversing its direction. Specifically, we transform each triple $(e_i, r_k, e_j)$ into $(e_j, r_k^{-1}, e_i)$, where $\mathcal{R}' = \{r^{-1} | r \in \mathcal{R}\}$, and define $\mathcal{T}'$ as set of inverse triples. We then propose the augmented knowledge graph $\mathcal{G}' = (\mathcal{E}, \mathcal{R}'', \mathcal{T}'')$, where $\mathcal{R}'' = \mathcal{R}' \cup \mathcal{R}$ and $\mathcal{T}'' = \mathcal{T}' \cup \mathcal{T}$.

### B. Triple representation learning

We present a triple representation learning method to make use of relations. For each triple $t_{ij}^k = (e_i, r_k, e_j) \in \mathcal{T}''$, it can be formulated as follows:

$$a_{ijk} = \mathbf{W_a} \cdot (h_i + h_j + g_k) \tag{5}$$

where $h_i$, $h_j$, $g_k \in \mathbb{R}^d$ are the initial embeddings of entities $e_i$, $e_j$ and relation $r_k$, respectively. $\mathbf{W_a} \in \mathbb{R}^{d \times d}$ is a linear transformation matrix.

After that, in order to learn and quantify the importance of triple $t_{ij}^k$, we introduce an attention vector $\mathbf{v_{att}}$ as below:

$$b_{ijk} = \text{LeakyReLU}(\mathbf{v_{att}} \cdot a_{ijk}) \tag{6}$$

where LeakyReLU is the non-linear activation function.

### C. representation learning of entities

Subsequently, we compute the attention value using the *softmax* function to derive the relative importance for entity $e_i$ as follows:

$$\alpha_{ijk} = \frac{exp(b_{ijk})}{\sum_{(r_r, e_n) \in \mathcal{N}_{e(i)}} exp(b_{inr})}, (r_k, r_r \in \mathcal{R}) \tag{7}$$

where $\mathcal{N}_{e(i)}$ denotes the neighborhoods of $e_i$ in $\mathcal{T}$. Similarly, we can calculate $\beta_{ijk}$ by replacing $\mathcal{R}$ and $\mathcal{T}$ with $\mathcal{R}'$ and $\mathcal{T}'$ respectively.

Therefore, to update the representation of entity $e_i$, we further categorize and aggregate the representations of these triples based on their unique relation types. The specific definitions as follows:

$$h'_{i\_O} = f_e\left(\mathbf{W}_{\text{O}} \sum_{(r_k, e_j) \in \mathcal{N}_{e(i)}} \alpha_{ijk} a_{ijk}\right), r_k \in \mathcal{R} \tag{8}$$

$$h'_{i\_I} = f_e\left(\mathbf{W}_{\text{I}} \sum_{(r_k, e_j) \in \mathcal{N}_{e(i)}} \beta_{ijk} a_{ijk}\right), r_k \in \mathcal{R}' \tag{9}$$

where $h'_{i\_O}$ and $h'_{i\_I}$ represent the representations of entity $e_i$ aggregated from the original and inverse relations, respectively. $f_e$ denotes a non-linear activation function such as $tanh$. $\mathbf{W}_{\text{O}}$, $\mathbf{W}_{\text{I}} \in \mathbb{R}^{d' \times d}$ are the graph convolutional kernels. To fuse the aggregated representations and incorporate residual connection, we employ the gating unit mentioned above. This is done as follows:

$$h'_i = gate(\mathbf{W}_{\text{e}} \cdot h_i, h'_{i\_O} \| h'_{i\_I}) \tag{10}$$

where $h'_i$ indicates the final representation of entity $e_i$. $\mathbf{W}_{\text{e}} \in \mathbb{R}^{d' \times d}$ is a linear transformation matrix.

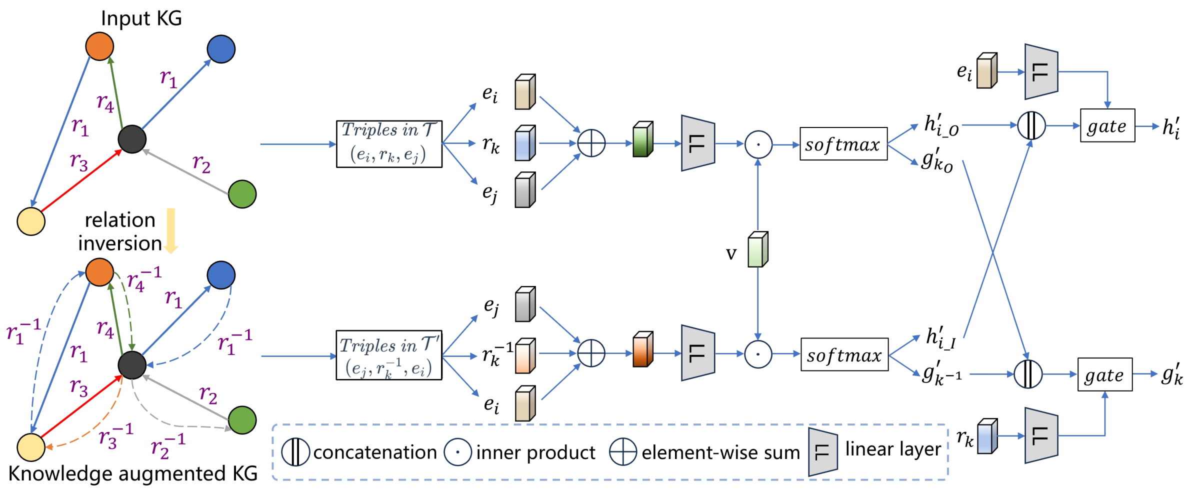


Fig. 1. Single attention head framework of our proposed model (KAAGE). $e_i$, $r_k$, $e_j$, $r_k^{-1}$ are initial embeddings of entity $i$, relation $k$, entity $j$, inverse relation $k^{-1}$, respectively. "$softmax$" represents the softmax function and $\mathbf{v}$ is an attention vector. "$gate$" denotes the gating unit which fuses information and $h_i'$, $g_k'$ are the final representations of entity $i$ and relation $k$ for each attention head.

Here, we adopt the multi-head attention mechanism, and the ultimate representation can be concatenated in the following formula:

$$h_i = \overset{M}{\underset{m=1}{||}} h'_{i_m} \tag{11}$$

where $M$ denotes the total number of attention heads and $h'_{i_m}$ signifies the final representation of $e_i$ within the $m$-th attention head.

### D. Representation learning of relations

Similarly, for each relation $r_k \in \mathcal{R}$, we compute its relative importance. The same process is followed for the inverse relation $r_k^{-1} \in \mathcal{R}'$:

$$\gamma_{ijk} = \frac{exp(b_{ijk})}{\sum_{(e_m,e_n)\in\mathcal{N}_{r(k)}} exp(b_{mnk})}, (k \in \mathcal{R}) \tag{12}$$

where $\mathcal{N}_{r(k)}$ denotes the set of related head-tail entity pairs of $r_k$. Also $\delta_{ijk^{-1}}$ can be obtained by replacing $\mathcal{N}_{r(k)}$ with $\mathcal{N}_{r(k^{-1})}$.

Then we proceed to aggregate the representations of the related triples based on these importance values:

$$g'_{k_O} = f_r\left(\mathbf{W}_{\mathrm{R}} \sum_{(e_i,e_j)\in\mathcal{N}_{r(k)}} \gamma_{ijk} a_{ijk}\right) \tag{13}$$

$$g'_{k^{-1}} = f_r\left(\mathbf{W}_{\mathrm{R}'} \sum_{(e_i,e_j)\in\mathcal{N}_{r(k^{-1})}} \delta_{ijk^{-1}} a_{ijk^{-1}}\right) \tag{14}$$

where $f_r$ represents a non-linear activation function of relations such as tanh. $\mathbf{W}_{\mathrm{R}}$ and $\mathbf{W}_{\mathrm{R}'}$ are the graph convolutional kernels.

Analogous to the approaches used for entities, we integrate a residual connection into the final representation and fuse it with a gating unit to enhance the model's performance:

$$g'_k = gate(\mathbf{W}_{\mathrm{r}} \cdot g_k, g'_{k_O} \| g'_{k^{-1}}) \tag{15}$$

where $\mathbf{W}_{\mathrm{r}}$ is a linear transformation matrix.

Ultimately, the combined representation of the relation $r_k$ with $M$ attention heads can be concatenated as described below:

$$g_k = \overset{M}{\underset{m=1}{||}} g'_{k_m} \tag{16}$$

### E. Training

***Score function.*** In our model, we utilize ConvE as the decoder to determine the plausibility scores for all potential triples. Specifically, as detailed in Equation 1, the score for the triple $t_{ij}^k$ can be calculated in the following formula:

$$\varphi(e_i, r_k, e_j) = f(vec(f([\overline{h_i}; \overline{g_k}] * w))\mathbf{W})h_j \tag{17}$$

***Loss function.*** The challenge of link prediction in KG can be framed as a binary classification task, where the goal is to distinguish between the presence and absence of a link. To achieve this, we adopt the binary cross-entropy loss function to train our model effectively.

$$\mathcal{L} = -\frac{1}{N}\sum_{o=1}^{N}\left(y_{t_{io}^k} log(\hat{y}_{t_{io}^k}) + (1 - y_{t_{io}^k}) log(1 - \hat{y}_{t_{io}^k})\right) \tag{18}$$

where $N$ is the number of candidates of tail entities. $y_{t_{io}^k} \in \{0, 1\}$ represents the real label of triple $t_{io}^k$. Employing sigmoid to the score, $\hat{y}_{t_{io}^k} = sigmoid(\varphi(e_i, r_k, e_o)) \in [0, 1]$ can be obtained.

## IV. Experiments and Analysis

### A. Datasets

In this study, we carefully selected three high-quality experimental datasets including **FB15K-237** [14], **WN18RR** [13], **Kinship** [11] to thoroughly assess and validate the efficacy and robustness of our proposed model. These three benchmark datasets encompass a range of relation and entity counts, with their fundamental statistical details neatly outlined in Table I.

TABLE I
DATASET STATISTICS

| Dataset | Entities | Relations | Triples | | | |
|---|---|---|---|---|---|---|
| | | | Train | Valid | Test | All |
| FB15k-237 | 14 541 | 237 | 272 115 | 17 535 | 20 466 | 310 116 |
| WN18RR | 40 943 | 11 | 86 835 | 3034 | 3134 | 93 003 |
| Kinship | 104 | 25 | 8544 | 1068 | 1074 | 10 686 |

TABLE II
PARTIAL HYPERPARAMETER SETTINGS FOR EACH DATASET

| Dataset | WN18RR | FB15k-237 | Kinship |
|---|---|---|---|
| Learning rate | 1e-4 | 1e-4 | 1e-4 |
| Epochs | 800 | 1000 | 1000 |
| Batch size | 128 | 128 | 128 |
| Embedding dimension | 200 | 300 | 300 |
| GCN dropout rate | 0.5 | 0.6 | 0.4 |
| Attention heads | 1 | 3 | 3 |

### B. Evaluation metrics

In this paper, we evaluate our model using the rankings of triples. We assess performance of our model with simple metrics such as Mean Reciprocal Rank (MRR), Mean Rank (MR), and Hits@N, all based on how well the model ranks the correct triples.

### C. Baselines and experimental settings

Here, we introduce a few baseline models tailored for the task of link prediction in KGs, to validate the effectiveness of our proposed model. These baselines are outlined as follows:

- TransE [6]: A translational model that represents entities and relations in vectors, minimizing the distance between them through translations to model relations.
- DistMult [15]: A tensor factorization model using bilinear scoring via matrix multiplication to represent entity-relation intricacies.
- ComplEx [16]: Extends DistMult to complex numbers, enabling capture of asymmetric and intricate relations between entities and relations.
- ConvE [13]: A CNN-based model applying convolutional operations to process relations and entities, inferring missing links through neural networks.
- WGCN [17]: A GCN variant modeling relation differences with weighted graph convolutions and learnable scalar weights for neighborhood messages.
- ComplexGCN [18]: Utilizes complex domain convolutions in GCNs to capture knowledge representations of entities and relations.
- MSHE [19]: a novel link prediction framework for knowledge graph embedding, leveraging multi-source and hierarchical neural networks to integrate complex knowledge.
- D-AEN [20]: A dual-attention embedding network for KGE, fusing neighborhood info with bidirectional and relation-specific attentions to propagate and update representations.

Our model implementation relies on the open-source *PyTorch* platform, executed on an Ubuntu 18.04 server powered by an *Intel Xeon Silver 4210R CPU* and an *NVIDIA GeForce RTX 3090 GPU*. The hyperparameters tailored for each dataset are detailed in Table II.

### D. Results and discussions

Table III showcases the impressive link prediction capabilities of our KAAGE model across three renowned benchmark datasets. By benchmarking against various baseline models, we confirm that KAAGE excels, particularly on the Kinship dataset.

Specifically, by observing the performance of these models, we can conclude that: Our model performs best on most evaluation metrics of the three datasets. On the FB15k-237 dataset, ComplexGCN has the best Hits@1 and the second-best MRR because ComplexGCN maps entities and relations to the complex space, which has better representation capabilities for the multi-type and general-domain knowledge graph dataset such as FB15k-237. However, our KAAGE is still optimal on the other three metrics. Similarly, D-AEN, which reverses relations and triples, does not use a gating mechanism when fusing heterogeneous information, but uses simple addition, resulting in slightly inferior performance.

## V. CONCLUSIONS

In this article, we introduce KAAGE, a novel GAT-based approach for link prediction. KAAGE augment knowledge by inversing relations and triples, leveraging attention mechanism to fuse triple representations, capturing rich semantics. It merges original and augmented knowledge via a gating unit, balancing structural information. Extensive experiments on three benchmarks reveal KAAGE's superiority, especially on Kinship.

## ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation of China (Grant No: 61901116, Grant No: 61471229 and Grant No:62001115), the Guangdong Basic and Applied Basic Research Foundation (Grant No: 2023A1515140111, Grant No: 2019A1515010789, Grant No: 2021A1515012289 and Grant No: 2019A1515110136) and the Jihua Laboratory Scienctific project (Grant No: X210101UZ210).

## REFERENCES

[1] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 1271–1279.

[2] Z. Liu, C. Xiong, M. Sun, and Z. Liu, "Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval," 2018.

TABLE III
HITS@1, HITS@3, HITS@10, MR, AND MRR RESULTS FOR LINK PREDICTION ON WN18RR, FB15K-237 AND KINSHIP. BOLD SCORES REPRESENT THE BEST RESULTS, AND UNDERLINED SCORES INDICATE THE SECOND-BEST RESULTS. "H@N" IS SHORT FOR "HITS@N"

| | WN18RR | | | | | FB15k-237 | | | | | Kinship | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Models | H@1 | H@3 | H@10 | MR | MRR | H@1 | H@3 | H@10 | MR | MRR | H@1 | H@3 | H@10 | MR | MRR |
| TransE | - | - | 0.501 | 3384 | 0.226 | - | - | 0.465 | 357 | 0.294 | 0.009 | 0.643 | 0.841 | 3.8 | 0.309 |
| DistMult | 0.39 | 0.44 | 0.49 | 5110 | 0.43 | 0.155 | 0.263 | 0.419 | 254 | 0.241 | 0.367 | 0.581 | 0.867 | 5.26 | 0.516 |
| ComplEx | 0.41 | 0.46 | 0.51 | 5261 | 0.44 | 0.158 | 0.275 | 0.428 | 339 | 0.247 | 0.733 | 0.899 | 0.971 | 2.48 | 0.823 |
| ConvE | 0.39 | 0.43 | 0.48 | 5277 | 0.46 | 0.239 | 0.350 | 0.491 | 246 | 0.316 | 0.73 | 0.91 | 0.98 | 2 | 0.83 |
| WGCN | 0.43 | 0.48 | 0.54 | - | 0.47 | 0.26 | 0.39 | 0.54 | - | 0.35 | - | - | - | - | - |
| ComplexGCN | 0.245 | 0.371 | 0.524 | - | 0.338 | **0.423** | 0.468 | 0.516 | - | 0.455 | - | - | - | - | - |
| MSHE | 0.43 | 0.48 | 0.54 | - | 0.47 | 0.26 | 0.39 | 0.54 | - | 0.35 | - | - | - | - | - |
| D-AEN | **0.443** | 0.500 | 0.561 | 2248 | 0.484 | 0.337 | 0.471 | 0.611 | 164 | 0.429 | 0.968 | 0.984 | **0.990** | **1.52** | 0.977 |
| KAAGE | 0.442 | **0.509** | **0.564** | **1969** | **0.487** | 0.365 | **0.500** | **0.634** | **145** | **0.456** | **0.980** | **0.986** | **0.990** | 1.65 | **0.983** |


[3] X. Dong, E. Gabrilovich, G. Heitz, *et al.*, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 601–610.

[4] Q. Liu, Y. Li, H. Duan, Y. Liu, and Z. Qin, "A survey on knowledge graph construction techniques," *Journal of Computer Research and Development*, vol. 53, pp. 582–600, 2016, ISSN: 1000-1239.

[5] Z. Liu, M. Sun, Y. Lin, and R. Xie, "Progress in knowledge representation learning," *Journal of Computer Research and Development*, vol. 53, pp. 247–261, 2016, ISSN: 1000-1239.

[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[7] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss markov random fields and probabilistic soft logic," *Journal of Machine Learning Research*, vol. 18, no. 109, pp. 1–67, 2017.

[8] Q. Wang, J. Liu, Y. Luo, B. Wang, and C.-Y. Lin, "Knowledge base completion via coupled path ranking," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1308–1318.

[9] M. Nickel, V. Tresp, H.-P. Kriegel, *et al.*, "A three-way model for collective learning on multi-relational data.," in *Icml*, vol. 11, 2011, pp. 3 104 482–3 104 584.

[10] L. Guo, Q. Zhang, W. Ge, W. Hu, and Y. Qu, "Dskg: A deep sequential model for knowledge graph completion," in *Knowledge Graph and Semantic Computing. Knowledge Computing and Language Understanding: Third China Conference, CCKS 2018, Tianjin, China, August 14–17, 2018, Revised Selected Papers*, Springer, 2019, pp. 65–77.

[11] X. V. Lin, R. Socher, and C. Xiong, "Multi-hop knowledge graph reasoning with reward shaping," *arXiv preprint arXiv:1808.10568*, 2018.

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

[13] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

[14] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, 2015, pp. 57–66.

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

[16] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*, PMLR, 2016, pp. 2071–2080.

[17] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 3060–3067.

[18] A. Zeb, S. Saif, J. Chen, A. U. Haq, Z. Gong, and D. Zhang, "Complex graph convolutional network for link prediction in knowledge graphs," *Expert Systems with Applications*, vol. 200, p. 116 796, 2022.

[19] D. Jiang, R. Wang, L. Xue, and J. Yang, "Multisource hierarchical neural network for knowledge graph embedding," *Expert Systems with Applications*, vol. 237, p. 121 446, 2024.

[20] H. Fang, Y. Wang, Z. Tian, and Y. Ye, "Learning knowledge graph embedding with a dual-attention embedding network," *Expert Systems with Applications*, vol. 212, p. 118 806, 2023.