

Article

Link Prediction Based on Feature Mapping and Bi-Directional Convolution

Ping Feng ^{1,2,3,4,5} , Xin Zhang ⁶, Hang Wu ⁶, Yunyi Wang ⁶, Ziqian Yang ⁶ and Dantong Ouyang ^{1,*} 

¹ College of Computer Science and Technology, Jilin University, Changchun 130012, China; fengping@ccu.edu.cn

² College of Computer Science and Technology, Changchun University, Changchun 130022, China

³ Ministry of Education Key Laboratory of Intelligent Rehabilitation and Barrier-Free Access for the Disabled, Changchun 130022, China

⁴ Jilin Provincial Key Laboratory of Human Health State Identification and Function Enhancement, Changchun 130022, China

⁵ Jilin Rehabilitation Equipment and Technology Engineering Research Center for the Disabled, Changchun 130022, China

⁶ College of Cybersecurity, Changchun University, Changchun 130022, China; zx_silence@163.com (X.Z.); wuhhang@163.com (H.W.); 17304352555@163.com (Y.W.); 19003063186@163.com (Z.Y.)

* Correspondence: dantong1206@163.com; Tel.: +86-13504313045

Abstract: A considerable amount of research on link prediction has recently been driven by missing relationships between knowledge graph entities and the problem of the incompleteness of knowledge graphs. Some recent studies have shown that convolutional neural networks based on knowledge embeddings are highly expressive and have good performance in link prediction. However, we found that the convolutional neural network (CNN)-based models do not handle the link between relations and entities well. For this reason, this paper proposes a link prediction model (LPM) based on feature mapping and bi-directional convolution. For the modeling of the task, an encoding layer–mapping layer–decoding layer structure is used. Among these layers, the encoding layer adopts a graph attention network to encode multi-hop triad information and obtains richer encoding of entities and relationships. The mapping layer can realize the mapping transformation between entities and relations and project the entity encoding in the space of relation encoding to capture the subtle connection between entities and relations. The decoding layer adopts bidirectional convolution to merge and decode the triples in a sequential inverse order, which makes the decoding layer model more advantageous in prediction. In addition, the decoding layer also adopts the r-drop training method to effectively reduce the distribution error generated by training between models and enhance the robustness of the model. Our experiments demonstrated the effectiveness of mapping relations, bidirectional convolution, and r-drop, and the accuracy of the proposed model showed significant improvements for each evaluation metric on two datasets, WN18RR and FB15k-237.

Keywords: knowledge graph; link prediction; knowledge embedding; feature mapping; convolutional neural network



Citation: Feng, P.; Zhang, X.; Wu, H.; Wang, Y.; Yang, Z.; Ouyang, D. Link Prediction Based on Feature Mapping and Bi-Directional Convolution. *Appl. Sci.* **2024**, *14*, 2089. <https://doi.org/10.3390/app14052089>

Academic Editor: Douglas O'Shaughnessy

Received: 25 January 2024

Revised: 24 February 2024

Accepted: 29 February 2024

Published: 2 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Knowledge graph [1] is a method for representing a knowledge base as a directed graph, where nodes and edges represent entities and relationships between entities, respectively. Knowledge graphs play an important role in practical applications in areas such as semantic search [2], dialog generation [3], question and answer (Q&A) [4], and recommendation algorithms [5]. However, knowledge graphs often face the problem of missing relations, i.e., there are unknown relations in the graph. To address this problem, knowledge base completion tasks [6], also known as link prediction, have emerged to predict whether a given triple is valid, e.g., (Yao Ming, gender, man).

Currently, models based on knowledge embedding and convolutional neural networks

have achieved satisfactory results in link prediction. However, the current knowledge embedding methods [7] do not take into account the subtle links between entities and relationships, although they contain multi-hop ternary information with high quality. Multi-hop ternary information is used to add multiple intermediate entities and relationships between the head entity, relationship, and tail entity; adding new entities and relationships can constitute more triads. In addition, existing convolutional neural network models mainly perform convolutional operations on the ternary directly while ignoring the sequential relationships between entities and relations in the ternary.

Based on the above observations, this paper proposes a link prediction model based on feature mapping and bidirectional convolution. (1) Firstly, the multi-hop ternary information is fed into the graph attention network to realize the encoding of entities and relations. (2) After encoding entities and relationships, the entities are mapped according to the space of relationship encoding to acquire the encoding of entities in the space of relationships, which can capture the links between entities and relationships. (3) After the mapped entity-relationship encoding is obtained, the ternary is combined in two sequences in reverse order, and the combined ternary is encoded and then fed into the convolutional neural network, which is trained by the r-drop method. The bi-directionally combined ternary will be more advantageous in predicting the head and tail entities after the information is extracted by the convolutional neural network. Our architecture model includes a coding layer, mapping layer, and decoding layer, in which the multi-hop graph neural coding layer, the entity relationship transformation layer, and the bidirectional convolutional layer play the roles of coding layer, mapping layer, and decoding layer, respectively.

The contributions of this paper are as follows. To the best of our knowledge, this paper introduces, for the first time, the mapping of entities and relations in a convolutional neural network for dealing specifically with the problem of link prediction on knowledge graphs. This mapping mechanism is the innovation of this paper. Secondly, we extend the convolutional neural network using bidirectional convolution to extract information from ternary groups. This bi-directional convolutional approach is important in the link prediction task. Finally, we evaluated our model in detail on two datasets, FB15K-237 and WN18R. We conducted corresponding experiments on both datasets, and the results demonstrate that the method proposed in this paper significantly outperforms the baseline model.

2. Related Work

Recent link prediction methods can be broadly categorized as follows: translation-based models, multiplication-based models, CNN-based models, rotation-based models, GNN-based models, and others.

2.1. Translation-Based Models

Bordes et al. [8] proposed the TransE model, which is modeled by interpreting triples as translations of entities over low-dimensional embeddings. Although TransE achieves good results, it cannot handle complex relations, such as self-inverse, one-to-many, many-to-one, and many-to-many cases. For this reason, Wang et al. [9] proposed the TransH model, which models a relation as a hyperplane and performs translation operations on it, which can well maintain the mapping nature of the relation. However, an entity may have more than one aspect, and various relationships may focus on different aspects of the entity, so Lin et al. [10] proposed the TransR model, which constructs entity and relationship embeddings in separate entity and relationship spaces and projects the features between entities to model the translation. Ji et al. [11] proposed a finer-grained TransD model, which uses two vectors to represent entities and relationships. Not only the diversity of relations but also the diversity of entities is considered. Fan et al. [12] proposed the TransM high-level model, which utilizes the structure of the knowledge graph to pre-compute different weights for each training triad based on its relation mapping attributes. In this way, the optimal function handles each ternary that depends on its weights, improving the overall performance of the model. Zhang et al. [13] proposed the TransAP model,

introducing location-aware embeddings and self-attention blocks which proved capable in adapting to various translational distance models. Lu et al. [7] proposed PTransE and AMIE+ to extract multi-hop and rule-based relationships.

2.2. Multiplication-Based Models

Nickel et al. [14] proposed the Rescal model, a tensor decomposition-based relational learning method. The model is able to collectively learn through potential components to improve the ability to judge ternary groups. Yang et al. [15] proposed the Distmult model, which captures the relational semantics using a bi-linear goal-learning approach. Trouillon et al. [16] used complex-valued embeddings and proposed the CompIEX model, where the combination of complex-valued embeddings can handle a large number of binary relations, including symmetric and anti-symmetric relations. Kazemi et al. [17] proposed the SimpleX model, which allows for independent learning of two embeddings per entity, with complexity that grows linearly with the size of the embeddings. Balažević et al. [18] proposed the TuckER model, which uses a purely linear model to deal with the ternary prediction problem and achieved relatively good results.

2.3. CNN-Based Models

Dettmers et al. [19] proposed the ConvE model, a multi-layer convolutional network model as a first attempt to deal with connectivity prediction problems using convolutional neural networks. Subsequently, Nguyen et al. [20] proposed the ConvKB model, which improves the model performance using convolutional neural networks to capture global relationships and transition features between entities and relationships in the knowledge base. Vu et al. [21] proposed an embedding model for CapsE, which employs a two-fold capsule network to model the ternary, generating an intermediate hidden layer through multiple filters to produce a continuous vector. Two convolutions are used to improve the performance of the model. Balažević et al. [22] proposed a hyper-network architecture HypER model that generates simplified relation-specific convolutional filters for relation prediction and also achieved relatively good results. On the other hand, Vashishth et al. [23] proposed the InteractE model, which employs feature alignment, new feature reshaping, and cyclic convolution, allowing for improved model performance.

2.4. Rotation-Based Models

Sun et al. [24] proposed a new knowledge graph embedding method, the RotatE model, where each relation is defined as a rotation from the source entity to the target entity in a complex vector space, which is capable of inferring and modeling a variety of relational patterns, and the embedding in the complex vector space also leads to a significant improvement in the performance of the model. Zhang et al. [25] proposed the QuatE model, which goes beyond the traditional complex-valued representation, introducing a more expressive hyper-complex representation for entities and relations embedded in knowledge graphs, and relations are modeled as rotations in quaternion space, which likewise optimizes the performance of the model. Gao et al. [26] proposed the Rotate3D model, which maps entities to 3D space and defines relations as rotations from head entities to tail entities, exploiting the non-exchangeable composite property, which naturally maintains the order of the relationship composite. Chen et al. [27] proposed the HittER model, which consists of two different Transformer blocks: the bottom extracts the features of each entity–relationship pair in the local neighborhood of the source entity, and the top aggregates the relationship information outputted from the bottom. The introduction of the Transformer also improved the model performance. Abboud et al. [28] proposed a BoXE model based on spatial Transformer embedding, embedding entities and relations in the form of hyper-rectangles. These hyper-rectangles spatially describe the basic logical attributes, the model can capture the information from the rich rule language, and the global reasoning approach also improves the model performance.

2.5. GNN-Based Models

Schlichtkrull et al. [29] introduced relational graph convolutional networks to deal with link prediction models specifically developed to deal with the multi-relational data characteristics of realistic knowledge bases. Li et al. [30] also fused graph neural networks with multi-layer perceptron with good results. Tan et al. [31] proposed knowledge-relational attention networks (KRAT) that utilize the context of the graph to project neighboring triples to different potential spaces simultaneously, capturing the subtle semantic information and importance of different contextual triples. Liang et al. [32] aggregated multimodal and graph structural information to generate the final embedding of the knowledge graph for knowledge graph completion. Niu et al. [33] introduced graph convolutional networks (GCNs) into the channel attention. The convolutional encodes the node structural features and represents them in matrix form to fully exploit the node feature information. Zamini et al. [34] learned entities and relations by embedding them in a low-dimensional vector space with a graph neural network.

2.6. Other Models

Zhang et al. [35] proposed the regularization method DURA model, which specifically deals with the existing tensor decomposition-based models and regularizes the tensor decomposition-based models, improving the performance of tensor decomposition models. Nathani et al. [36] proposed the attention feature embedding method, which captures entity and relationship features in the neighborhood of any given entity with significant performance improvement. Chen et al. [37], on the other hand, proposed a meta-relational learning MetaR framework for the link prediction problem with few samples. Baek et al. [38] proposed a meta-learning-based framework GEN model for this problem. The model is used for inductive inference for node embedding networks, and for transformational inference for link prediction networks from the unseen entities in link prediction, the uncertainty is modeled, which also improves the accuracy of the model. Bose et al. [39] proposed a meta-learning approach for link prediction with fewer samples, where in addition to the global parameters, the model learns a graph signature function, which exploits the structural information of the graph as compared to other graphs from the same distribution, resulting in faster adaptation and better convergence than ordinary meta-learning. Xu et al. [40] proposed a cross-modal attention mechanism that provides bi-directional attention to visual and textual features by integrating relational embedding, thus improving the robustness of the model. Zhang et al. [41] introduced federated learning to the task of knowledge graph completion. Shomer et al. [42] introduced a data augmentation approach for knowledge graph completion. Luo et al. [43] used the sample-less learning approach for the task of knowledge graph completion. Wang et al. [44] proposed a multi-concept representation learning approach where each entity or relationship is projected to multiple vectors to capture the complex conceptual information hidden in them for optimal embedding.

3. Model Details

The experimental model in this paper is divided into three layers, starting with a coding layer for extracting coded representations of entities and relationships from the input data. The goal of the encoding layer is to capture the important features of the input data. Second is the mapping layer, through which entities are mapped in the relation space. The role of this layer is to model and represent the associations of entities across relationships. Finally, there is the decoding layer, which is used to decode the triples. The decoding layer reconverts the encoded entities and relations into their original ternary form for the link prediction task.

The encoding layer, as shown in Figure 1a, fuses the single-hop ternary information with the multi-hop ternary information, and the fused information is used to obtain the entity coding and relationship coding through the graph neural network; the mapping layer, as shown in Figure 1b, because the entity coding and relationship coding obtained

from the coding layer belong to different coding spaces, maps the entity coding in the relationship coding space and obtains new entity coding, which strengthens the subtle connection between the entity and the relationship coding. The decoding layer, as shown in Figure 1c, strengthens the prediction capability of the model by combining the sequential and inverse sequence of the ternary information.

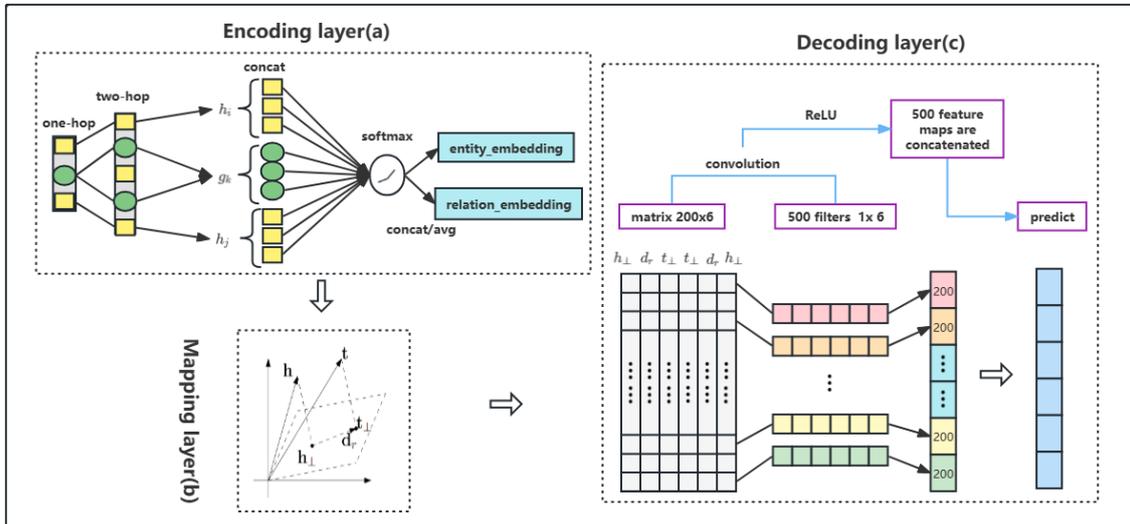


Figure 1. Overall model diagram divided into three parts: (a) encoding layer, (b) mapping layer, and (c) decoding layer.

3.1. Encoding Layer

In order to obtain the embedding of entities and relations, it is necessary to learn the representation of each ternary. We have the entity, and relation feature vectors of the single-hop triad $t_{i_s j_s}^{m_s} = (e_{i_s}, r_{m_s}, e_{j_s})$ and the multi-hop triad $t_{i_d j_d}^{m_d n_d} = (e_{i_d}, r_{m_d}, e_{j_d}, r_{n_d}, e_{k_d})$ are concatenated to obtain a richer ternary representation. Multi-hop ternary can carry more information between entities and relations, which will only make the embedding richer. Firstly, the single-hop and multi-hop ternary are merged, where t denotes the ternary, s denotes the single-hop information, d denotes the multi-hop information, $e_{i_s}, e_{j_s}, e_{i_d}, e_{j_d}, e_{k_d}$ denote entity embeddings, and $r_{m_s}, r_{m_d}, r_{n_d}$ denote relation embeddings. First, the single-hop and multi-hop triples are merged as follows.

$$h_i = \text{concat}[e_{i_s}, e_{i_d}] \tag{1}$$

$$h_j = \text{concat}[e_{j_s}, e_{k_d}] \tag{2}$$

$$g_k = \text{concat}[r_{m_s}, r_{m_d}, r_{n_d}] \tag{3}$$

The linear transformation is then performed with the following equation.

$$c_{ijk} = [h_i || g_k || h_j] \tag{4}$$

where c_{ijk} is the vector representation of the triplet after linear transformation, the matrix h_i represents the embedding of the entities e_{i_s}, e_{i_d} , the matrix h_j represents the embedding of the entities e_{j_s}, e_{k_d} , and the matrix g_k represents the embedding of the relations $r_{m_s}, r_{m_d}, r_{n_d}$.

We perform a linear transformation parameterized by the weight matrix W_1 and then operate on it via the activation function LeakyRelu to obtain the absolute attention b_{ijk} of the ternary with the following formula.

$$b_{ijk} = \text{LeakyReLU}(W_1 c_{ijk}) \tag{5}$$

To obtain the relative attention α_{ijk} , the activation function softmax is applied to the following equation.

$$\alpha_{ijk} = \text{Softmax}_{jk}(b_{ijk}) \tag{6}$$

The embedding h'_i of entity e_i is the sum of each ternary representation weighted by their attention values, as shown in the following equation.

$$h'_i = \sigma\left(\sum_{j \in N_i} \sum_{k \in R_{ij}} \alpha_{ijk} c_{ijk}\right) \tag{7}$$

where N_i denotes the neighborhood of entity e_i and R_{ij} denotes the set of relations connecting entity e_i and e_j . To stabilize the learning process and encapsulate more information about the neighborhood, the attention mechanism calculates the embedding and then connects the embeddings with the following equation.

$$h'_i = \sum_{m=1}^M \sigma\left(\sum_{j \in N_i} \sum_{k \in R_{ij}} \alpha_{ijk}^m c_{ijk}^m\right) \tag{8}$$

In the last layer of the model, rather than joining the embeddings of multiple heads, an averaging method is used to obtain the final embedding vector of the entity, as shown in the following equation.

$$h'_i = \sigma\left(\frac{1}{M} \sum_{m=1}^M \sum_{j \in N_i} \sum_{k \in R_{ij}} \alpha_{ijk}^m c_{ijk}^m\right) \tag{9}$$

However, when learning new embeddings, entities lose their initial embedding information. To solve this problem, we use a matrix vector W^E to perform a linear transformation with H^i , where H^i represents the input entity embeddings of our model, and then add H^f to the final encoding, which represents the entity information of the initial embeddings, to obtain the result of the linear transformation, H'' , as shown in the following equation.

$$H'' = W^E H^i + H^f \tag{10}$$

The experiments use hinge loss to train the model, and the hinge loss is given by the following expression.

$$L(\Omega) = \sum_{t_{ij} \in S} \sum_{t'_{ij} \in S'} \max\{d_{t'_{ij}} - d_{t_{ij}} + \gamma, 0\} \tag{11}$$

where $\gamma > 0$ is a hyperparameter, S is the set of valid triples, and S' denotes the set of invalid triples. In our architecture, we extend edges to directed paths by introducing an auxiliary relation between two entities. The embedding of this auxiliary relation is the sum of all relation embeddings in the path. Our model iteratively accumulates knowledge from the multi-hop information of the entities.

3.2. Mapping Layer

For relation r , we locate the relation-specific translation vector d_r in the relation-specific hyperplane w_r instead of in the same space where the entities are embedded so that the entity and relation distributions use different coding spaces, and then the projections are used to match the relations and the entities to allow for a more diversified coding of the entities and relations. Specifically, for a triple $t_{ij}^k = (e_i, r_k, e_j)$, the embedded head entity h and tail entity t are first projected to the hyperplane w_r with projections h_{\perp} and t_{\perp} , respectively. It is expected that h_{\perp} and t_{\perp} can be associated with the translation vector d_r in the hyperplane if (e_i, r_k, e_j) is the correct triple, and the projection function is shown below.

$$h_{\perp} = h - w_r^{\top} h w_r \tag{12}$$

$$t_{\perp} = t - w_{\gamma}^{\top} t w_r \quad (13)$$

3.3. Decoding Layer

In our experiments, we employed bi-directional convolution as part of the decoder, aiming to analyze the features of global embedding and bi-directional coding. By employing bidirectional coding, we give the model a stronger edge in predicting entities. Subsequent experimental results demonstrated that bi-directional coding shows good results in improving the performance of the model, and the scoring function of the decoding layer is shown by the following equation.

$$h_{triple} = \text{concat}[h_{\perp}, d_r, t_{\perp}, t_{\perp}, d_r, t_{\perp}, h_{\perp}] \quad (14)$$

$$f(t_{ij}^k) = (\|_{m=1}^{\Omega} \text{ReLU}(h_{triple} * w^m))W \quad (15)$$

where w^m denotes the convolution filter, Ω denotes the hyperparameter of the number of filters used, $*$ is the convolution operation, and W denotes the linear transformation matrix used to calculate the final score of the triple.

The r-drop training method is used while decoding due to the randomness of the dropout, and although the same model is passed twice, it is also slightly different in the forward passing process. r-drop makes each batch of data pass through the forward neural network before and after two times to acquire two different distributions. The KL dispersion between the two distributions generated by the r-drop is continuously reduced during the training process, which in turn improves the robustness of the model. The loss function used in the model is shown below.

$$L = \sum_{t_{ij}^k \in \{S \cup S'\}} \log(1 + \exp(L_{t_{ij}^k} \cdot f(t_{ij}^k))) + \frac{\lambda}{2} \|W\|_2^2 \quad (16)$$

$$L_{t_{ij}^k} = \begin{cases} 1 & \text{for } t_{ij}^k \in S \\ -1 & \text{for } t_{ij}^k \in S' \end{cases} \quad (17)$$

where L is the loss function of the decoding layer; L consists of the softplus function and the L_2 regularization, where the coefficient of the L_2 regularization is $\lambda/2$; and S is the triple of positive samples while S' is the triple of negative samples.

4. Experiment and Analysis

4.1. Dataset

We evaluated our proposed method using two benchmark datasets, WN18RR and FB15k-237. WN18RR (WordNet-18 Reversed Relations) is a commonly used knowledge graph link prediction dataset for evaluating and comparing the performance of link prediction models. This dataset is an extension of the WordNet-18 dataset, an English vocabulary network representing relationships between words, with 40,943 entities and 11 relationships. The FB15k-237 dataset was pruned with respect to the original FB15k dataset by removing some of the frequently occurring and relatively easy to predict relationships. This was done to increase the difficulty of the dataset, making it more accurate to assess the performance of the model in dealing with complex and rare relationships, with a total of 14,541 entities and 237 relationships in the dataset. Other specific data are shown in Table 1.

Table 1. Details of the dataset.

Dataset	Entities	Relations	Training	Validation	Test	Total
WN18RR	40,943	11	86,835	3034	3134	93,003
FB15k-237	14,541	237	272,115	17,535	20,466	310,116

4.2. Training

Experiments were conducted to partially augment the data by creating two sets of invalid triples, one set replacing the head entities in the triples with invalid entities and the other set replacing the tail entities in the triples with invalid entities. We randomly drew the same number of invalid triples from these two sets to ensure robust performance in detecting head and tail entities. The experimental initial embedding used TransE-generated entity and relation embeddings to initialize our embedding.

The experiment adopted the training process of encoder–mapper–decoder. Firstly, the multi-hop triple information was fed into the graph attention network for encoding, and the encoding of entities and relations of the knowledge graph was obtained. Then, the entities were mapped according to the space of relations, and the mapped entity–relationship encoding was obtained. Finally, the bi-directional convolutional decoder was trained to perform the relation prediction task.

Different parameter settings were used for different dataset experiments. (1) For the WN18RR dataset, we used Adam to optimize all parameters, the coding layer epoch was 3600, the initial learning rate of the coding layer was set to 5×10^{-6} , the decoding layer epoch was 200, the initial learning rate of the decoding layer was set to 1×10^{-5} , the decoding layer *batch_size* was 64, and both entity, and the relation embedding of the last layer was set to 200. (2) For the FB15k-237 dataset, all parameters were also optimized using Adam, the epoch of the coding layer was 3000, the initial learning rate of the coding layer was set to 0.00001, the epoch of the decoding layer was 200, the initial learning rate of the decoding layer was set to 0.000001, the *batch_size* of the decoding layer was 64, and both the entity and relational embeddings of the last layer were set to 200.

4.3. Validation

In the link prediction task, the goal is to predict a triple; that is, to give the head entity as well as the relationship prediction tail entity or to give the head entity as well as the relationship prediction tail entity. We generated a set of (N-1) incorrect triples for each entity, replaced them with each other entity, and then assigned a value to each such triple. Subsequently, we sorted these scores in ascending order to obtain a ranking of correct triples (e_i, r_k, e_j) . The experiment filtered all the duplicates in the setup, i.e., when ranking, we removed the triples that already existed in the training, validation, or test set. The whole process was repeated by replacing the tail entities and reporting the average metric. We report the mean inverse rank (MRR), mean rank (MR), and hit rate (Hits@N), e.g., N = 1, 3, and 10 for the first hit, third hit, and tenth hit, respectively.

4.4. Model Comparison

The proposed model, denoted as ConvFM, was compared with the previously methods: TransE, ConvE, ConvKB, DisMuit, ComplEX, and KB-GAT.

4.4.1. Comparison with the Translation Model Approach

TransE models the entities and relations of a triad by embedding them in low dimensions, but the initial embedding information of the translation model is not rich enough, and there is no mapping of entities to the relational space to extract the connections between entity relations. In contrast, the ConvFM model not only has the mapping of entities to the relationship space but also has a richer encoding of entity relationships.

4.4.2. Comparison with Multiplication Methods

DisMuit uses a bi-linear objective learning approach to capture relational semantics, and ComplEX models entities and relations using complex-valued embedding, but tensor decomposition models are generally prone to overfitting and have poor overall results. The ConvFM model, on the other hand, is based on convolutional neural networks and has better overall results.

4.4.3. Comparison with CNN Methods

ConvE uses a multi-layer convolutional network model for link prediction of triples, and ConvKB uses a convolutional neural network to capture the global relationships and transition features between entities and relationships in the knowledge base, but none of the above models takes into account the order problem among the triples themselves. The ConvFM model, on the other hand, takes into account the sequential inverse order problem of the triads to improve the performance of the model.

4.4.4. Comparison with the KB-GAT Model

Although the KB-GAT model has a higher quality of knowledge embedding and uses convolutional neural network as the base model, it does not consider the mapping problem between entity and relational relations and ignores the connections between entity relations. The ConvFM model, on the other hand, obtains better results by mapping the entities to the relational space and obtaining the subtle connections between entity relations.

The prediction results of the model for both datasets are displayed in Table 2. These results clearly show that our proposed method achieved essentially the best results on both datasets, FB15k-237 and WN18RR.

Table 2. Different models for link prediction, with the best results highlighted in bold.

Model	WN18RR					FB15K-237				
	MR	MRR	@1	@3	@10	MR	MRR	@1	@3	@10
TransE	2300	0.243	4.27	44.1	53.2	323	0.279	19.8	37.6	44.1
ConvE	4464	0.456	41.9	47	53.1	245	0.312	22.5	34.1	49.7
ConvKB	1295	0.265	5.82	44.5	55.8	216	0.289	19.8	32.4	47.1
DisMuit	7000	0.444	41.2	47	50.4	512	0.281	19.9	30.1	44.6
CompIEX	7882	0.449	40.9	46.9	53	546	0.278	19.4	29.7	45
KB-GAT	1940	0.440	36.1	48.3	58.1	210	0.518	46	54	62.6
ConvFM	1421	0.475	40.2	51.1	60.7	126	0.515	45.2	55.6	66.6

5. Ablation Experiments

Given the strong experimental results of the proposed method, a closer examination of the specific contributions of its components is warranted. Knowledge embedding of multi-hop graph attention and one-way convolution are used as the basis of the model, and two-way convolutional networks, feature mapping, and R-drop were used as variables to set up controlled experiments.

First, in the experiments, all the triples were encoded through the coding layer to acquire the coded representation of entities and relationships. Next, experiments were conducted on two datasets by adding bi-directional convolution and feature mapping, and evaluation metrics such as Hits@1, Hits@3, Hits@10, and MRR were observed. The results showed that the model tends to stabilize after about 200 rounds of training. Thereafter, a control experiment was conducted, and the MR evaluation metrics of the three models on two datasets were observed. The experimental results show that after about 200 rounds of training, the models basically converge and achieve the optimal values among the five metrics.

This experiment used the coding result of the last epoch of the coding layer as the coded representation of entities and relations. After the feature mapping layer, the entities were mapped into the relationship space. In the decoding layer, the data of each evaluation index were counted every ten rounds. The results in Figure 2 show that after 200 rounds of training, the three models basically converge. Figures 2–4 show the specific results of the three models on the two datasets except for the MR evaluation metrics. Figure 5 shows the specific results of the three models on the two datasets for the MR evaluation metrics. The results show that the five metrics achieved optimal values at close to about 200 epochs.

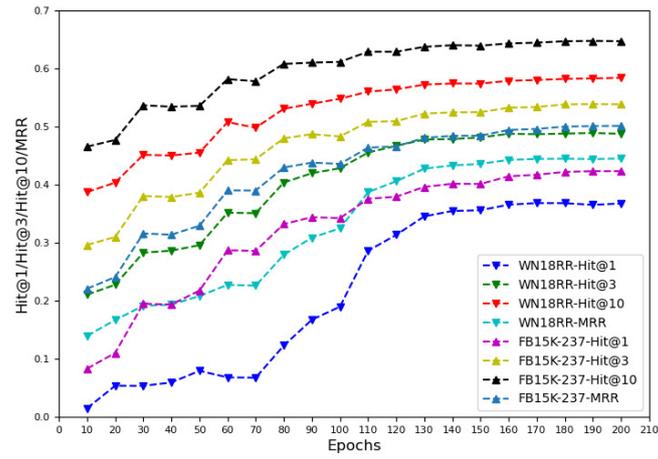


Figure 2. Results of ConGAT-BI.

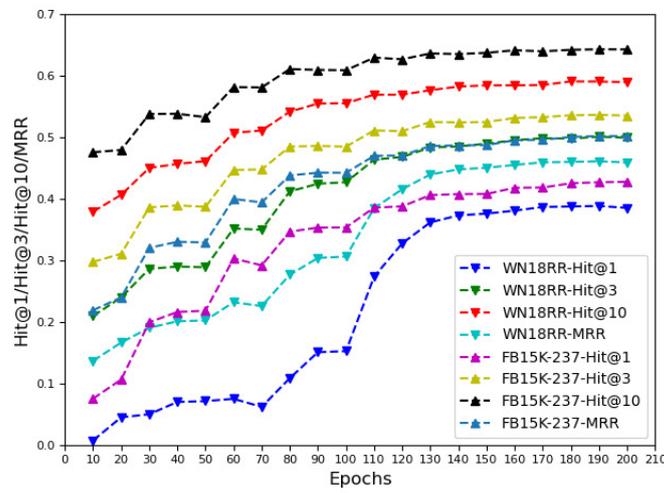


Figure 3. Results of ConGAT-BI-FM.

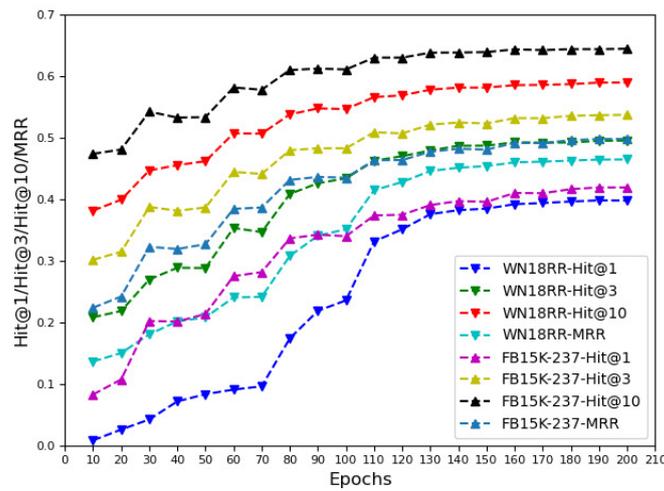


Figure 4. Results of ConGAT-BI-FM-rDrop.

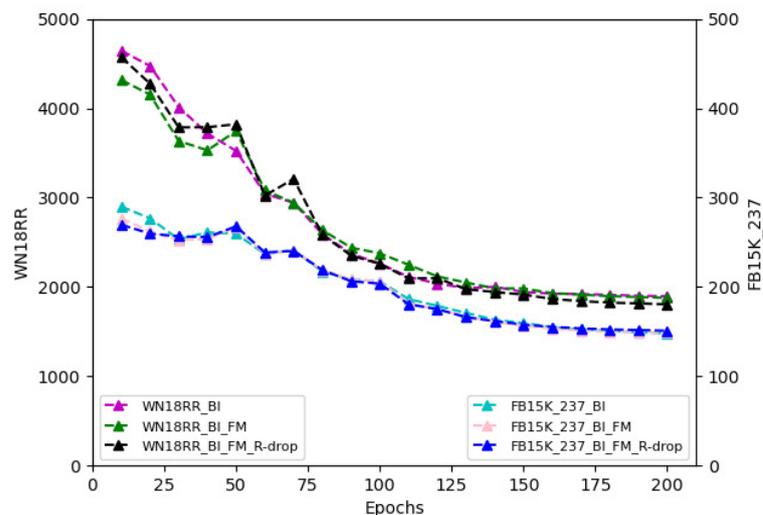


Figure 5. Results of MR.

When facing sparse and non-overlapping relations, our model was also able to show excellent performance. Multi-hop graph attention neural networks can provide significant advantages when dealing with these relations via the model's mapping mechanism. Graph neural networks are particularly good at aggregating sparse relational links in the knowledge graph through a sparse representation of the adjacency matrix. In addition, through the mapping layer of the model, we can effectively identify and process non-overlapping relations and realize accurate mapping. From now on, we will also focus on conducting detailed experiments on sparse or non-overlapping relation processing to further validate our conclusions and continuously improve the performance of the model in these areas.

The experiments used the coding layer of the graph attention network and the one-way convolution as the basis of the overall model (ConGAT). In the case of the basic convergence of the model, the models of all three control experiments performed well, as shown by Table 3, where the bi-directional convolution, feature mapping, and rDrop all led to different degrees of model improvement. The final experiment showed that the model ConGAT-BI-FM-rDrop achieved the best results.

Table 3. The specific results of the ablation experiments, with the best results highlighted in bold.

Model	WN18RR					FB15K-237				
	MR	MRR	@1	@3	@10	MR	MRR	@1	@3	@10
ConGAT-BI	2322	0.451	38.4	49.5	58.9	173	0.514	43.2	54.7	65.3
ConGAT-BI-FM	1451	0.470	39.2	50.6	59.6	148	0.515	43.8	55.2	65.9
ConGAT-BI-FM-rDrop	1421	0.475	40.2	51.1	60.7	126	0.515	45.2	55.6	66.6

6. Conclusions

This paper presents a link prediction model based on feature mapping and bidirectional convolution. The model first splices multi-hop and single-hop information and encodes it through a graph attention network to obtain an encoded representation of entities and relationships. Then, the model maps entities into the relationship coding space. Finally, the model splices the forward and inverse order triples and decodes them using a convolutional neural network, which is combined with r-drop for training. The model achieved good results on the WN18RR and FB15k-237 datasets.

However, the model has some limitations. First, the model uses a bi-directional convolutional and attentional mechanism during encoding and decoding, which may be limited by the localization of information transfer. Second, the model does not consider the encoding of multi-hop graph attention networks and the decoding of graph neural networks, which may limit the expressive power of the model. Therefore, in further research, the use of graph neural networks in the encoding and decoding process could be explored to

improve the performance of the model.

Future research directions could consider using graph neural networks for decoding based on feature mapping and multi-hop graph attention network encoding to achieve consistency in encoding and decoding models. This combined use of the same network structure may improve the representation and generalization capabilities of the model and further enhance the performance of the link prediction task. In addition, other improvements could be explored, such as introducing more sophisticated attention mechanisms or using more advanced pre-trained models to enhance model performance. These research directions are expected to promote the development and application of link prediction models in practical applications.

Author Contributions: Conceptualization, P.F. and D.O.; methodology, P.F. and X.Z.; software, H.W., Y.W. and Z.Y.; validation, P.F., D.O. and X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, P.F. and X.Z.; visualization, P.F. and X.Z.; funding acquisition, P.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Development Plan Project of Jilin Provincial Science and Technology Department (Key Technology Research on Risk Prediction and Assessment of Old Chronic Diseases Based on Medical Knowledge Graph (2023JB405L07)).

Data Availability Statement: Publicly available datasets were used in this study. This data can be found here (accessed on 1 March 2024): <https://huggingface.co/datasets/VLyb/WN18RR>; <https://huggingface.co/datasets/VLyb/FB15k-237>.

Acknowledgments: We would like to express our deepest gratitude to all those who have contributed to the completion of this research and the writing of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	convolutional neural network
LPM	link prediction model
KRAT	knowledge-relational attention network
GCN	graph convolutional network

References

- Peng, C.; Xia, F.; Naseriparsa, M.; Osborne, F. Knowledge graphs: Opportunities and challenges. *Artif. Intell. Rev.* **2023**, *56*, 13071–13102. [[CrossRef](#)]
- Muennighoff, N. Sgpt: Gpt sentence embeddings for semantic search. *arXiv* **2022**, arXiv:2202.08904.
- Kann, K.; Ebrahimi, A.; Koh, J.; Dudy, S.; Roncone, A. Open-domain dialogue generation: What we can do, cannot do, and should do next. In Proceedings of the 4th Workshop on NLP for Conversational AI, Dublin, Ireland, 27 May 2022; pp. 148–165.
- Du, H.; Le, Z.; Wang, H.; Chen, Y.; Yu, J. COKG-QA: Multi-hop question answering over COVID-19 knowledge graphs. *Data Intell.* **2022**, *4*, 471–492. [[CrossRef](#)]
- Srba, I.; Moro, R.; Tomlein, M.; Pecher, B.; Simko, J.; Stefancova, E.; Kompan, M.; Hrcakova, A.; Podrouzek, J.; Gavornik, A.; et al. Auditing YouTube’s recommendation algorithm for misinformation filter bubbles. *ACM Trans. Recommender Syst.* **2023**, *1*, 6. [[CrossRef](#)]
- Chen, Y.; Li, H.; Li, H.; Liu, W.; Wu, Y.; Huang, Q.; Wan, S. An overview of knowledge graph reasoning: Key technologies and applications. *J. Sens. Actuator Netw.* **2022**, *11*, 78. [[CrossRef](#)]
- Lu, X.; Wang, L.; Jiang, Z.; Liu, S.; Lin, J. MRE: A translational knowledge graph completion model based on multiple relation embedding. *Math. Biosci. Eng.* **2023**, *20*, 5881–5900. [[CrossRef](#)] [[PubMed](#)]
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–9. <https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf> (accessed on 28 February 2024).
- Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
- Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.

11. Ji, G.; He, S.; Xu, L.; Liu, K.; 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), Beijing, China, 26–31 July 2015; pp. 687–696.
12. Fan, M.; Zhou, Q.; Chang, E.; Zheng, F. Transition-based knowledge graph embedding with relational mapping properties. In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing, Phuket Island, Thailand, 12–14 December 2014; pp. 328–337.
13. Zhang, S.; Sun, Z.; Zhang, W. Improve the translational distance models for knowledge graph embedding. *J. Intell. Inf. Syst.* **2020**, *55*, 445–467. [[CrossRef](#)]
14. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the ICML, Bellevue, WA, USA, 28 June–2 July 2011; Volume 11, pp. 3104482–3104584.
15. Yang, B.; Yih, W.t.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
16. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning. PMLR, New York, NY, USA, 20–22 June 2016; pp. 2071–2080.
17. Kazemi, S.M.; Poole, D. Simple embedding for link prediction in knowledge graphs. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1–12. https://proceedings.neurips.cc/paper_files/paper/2018/file/b2ab001909a8a6f04b51920306046ce5-Paper.pdf (accessed on 28 February 2024).
18. Balažević, I.; Allen, C.; Hospedales, T.M. Tucker: Tensor factorization for knowledge graph completion. *arXiv* **2019**, arXiv:1901.09590.
19. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
20. Nguyen, D.Q.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv* **2017**, arXiv:1712.02121.
21. Vu, T.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A capsule network-based embedding model for knowledge graph completion and search personalization. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 3–5 June 2019; pp. 2180–2189.
22. Balažević, I.; Allen, C.; Hospedales, T.M. Hypernetwork knowledge graph embeddings. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; Proceedings 28; Springer: Cham, Switzerland, 2019; pp. 553–565.
23. Vashishth, S.; Sanyal, S.; Nitin, V.; Agrawal, N.; Talukdar, P. Interact: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3009–3016.
24. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv* **2019**, arXiv:1902.10197.
25. Zhang, S.; Tay, Y.; Yao, L.; Liu, Q. Quaternion knowledge graph embeddings. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11. https://proceedings.neurips.cc/paper_files/paper/2019/file/d961e9f236177d65d21100592edb0769-Paper.pdf (accessed on 28 February 2024).
26. Gao, C.; Sun, C.; Shan, L.; Lin, L.; Wang, M. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020; pp. 385–394.
27. Chen, S.; Liu, X.; Gao, J.; Jiao, J.; Zhang, R.; Ji, Y. Hitter: Hierarchical transformers for knowledge graph embeddings. *arXiv* **2020**, arXiv:2008.12813.
28. Abboud, R.; Ceylan, I.; Lukasiewicz, T.; Salvatori, T. Boxe: A box embedding model for knowledge base completion. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9649–9661.
29. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, 3–7 June 2018; Proceedings 15; Springer: Cham, Switzerland, 2018; pp. 593–607.
30. Li, J.; Shomer, H.; Ding, J.; Wang, Y.; Ma, Y.; Shah, N.; Tang, J.; Yin, D. Are Message Passing Neural Networks Really Helpful for Knowledge Graph Completion? In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 10696–10711.
31. Tan, Z.; Chen, Z.; Feng, S.; Zhang, Q.; Zheng, Q.; Li, J.; Luo, M. KRACL: Contrastive learning with graph context modeling for sparse knowledge graph completion. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–1 May 2023; pp. 2548–2559.
32. Liang, S.; Zhu, A.; Zhang, J.; Shao, J. Hyper-node relational graph attention network for multi-modal knowledge graph completion. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 62. [[CrossRef](#)]
33. Niu, H.; He, H.; Feng, J.; Nie, J.; Zhang, Y.; Ren, J. Knowledge graph completion based on GCN of multi-information fusion and high-dimensional structure analysis weight. *Chin. J. Electron.* **2022**, *31*, 387–396. [[CrossRef](#)]
34. Zamini, M.; Reza, H.; Rabiei, M. A review of knowledge graph completion. *Information* **2022**, *13*, 396. [[CrossRef](#)]

35. Zhang, Z.; Cai, J.; Wang, J. Duality-induced regularizer for tensor factorization based knowledge graph completion. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21604–21615.
36. Nathani, D.; Chauhan, J.; Sharma, C.; Kaul, M. Learning attention-based embeddings for relation prediction in knowledge graphs. *arXiv* **2019**, arXiv:1906.01195.
37. Chen, M.; Zhang, W.; Zhang, W.; Chen, Q.; Chen, H. Meta relational learning for few-shot link prediction in knowledge graphs. *arXiv* **2019**, arXiv:1909.01515.
38. Baek, J.; Lee, D.B.; Hwang, S.J. Learning to extrapolate knowledge: Transductive few-shot out-of-graph link prediction. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 546–560.
39. Bose, A.J.; Jain, A.; Molino, P.; Hamilton, W.L. Meta-graph: Few shot link prediction via meta learning. *arXiv* **2019**, arXiv:1912.09867.
40. Xu, D.; Xu, T.; Wu, S.; Zhou, J.; Chen, E. Relation-enhanced negative sampling for multimodal knowledge graph completion. In Proceedings of the 30th ACM International Conference on Multimedia, Lisbon, Portugal, 10–14 October 2022; pp. 3857–3866.
41. Zhang, K.; Wang, Y.; Wang, H.; Huang, L.; Yang, C.; Chen, X.; Sun, L. Efficient federated learning on knowledge graphs via privacy-preserving relation embedding aggregation. *arXiv* **2022**, arXiv:2203.09553.
42. Shomer, H.; Jin, W.; Wang, W.; Tang, J. Toward Degree Bias in Embedding-Based Knowledge Graph Completion. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–1 May 2023; pp. 705–715.
43. Luo, L.; Li, Y.F.; Haffari, G.; Pan, S. Normalizing flow-based neural process for few-shot knowledge graph completion. *arXiv* **2023**, arXiv:2304.08183.
44. Wang, J.; Wang, B.; Gao, J.; Hu, Y.; Yin, B. Multi-concept representation learning for knowledge graph completion. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 11. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.