Mathematical Biosciences and Engineering

MBE, 21(1): 1394–1412.
DOI: 10.3934/mbe.2024060
Received: 28 July 2023
Revised: 22 October 2023
Accepted: 13 November 2023
Published: 27 December 2023

http://www.aimspress.com/journal/MBE

Research article

Knowledge graph completion method for hydraulic engineering coupled with spatial transformation and an attention mechanism

Yang Liu[1,*], Tianran Tao[2], Xuemei Liu[2], Jiayun Tian[2], Zehong Ren[2], Yize Wang[2], Xingzhi Wang[2] and Ying Gao[3]

[1] Provincial Collaborative Innovation Center for Efficient Utilization of Water Resources in the Yellow River Basin, North China University of Water Resources and Electric Power, Zhengzhou 450046, China
[2] School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China
[3] Henan Water & Power Engineering Consulting CO., Ltd, Zhengzhou 450046, China

* **Correspondence:** Email: ly_research@126.com.

Abstract: In response to the limited capability of extracting semantic information in knowledge graph completion methods, we propose a model that combines spatial transformation and attention mechanisms (STAM) for knowledge graph embedding. Firstly, spatial transformation is applied to reorganize entity embeddings and relation embeddings, enabling increased interaction between entities and relations while preserving shallow information. Next, a two-dimensional convolutional neural network is utilized to extract complex latent information among entity relations. Simultaneously, a multi-scale channel attention mechanism is constructed to enhance the capture of local detailed features and global semantic features. Finally, the surface-level shallow information and latent information are fused to obtain feature embeddings with richer semantic expression. The link prediction results on the public datasets WN18RR, FB15K237 and Kinship demonstrate that STAM achieved improvements of 8.8%, 10.5% and 6.9% in the mean reciprocal rank (MRR) evaluation metric compared to ConvE, for the respective datasets. Furthermore, in the link prediction experiments on the hydraulic engineering dataset, STAM achieves better experimental results in terms of MRR, Hits@1, Hits@3 and Hits@10 evaluation metrics, demonstrating the effectiveness of the model in the task of hydraulic engineering knowledge graph completion.

Keywords: hydraulic engineering knowledge graph; knowledge graph completion; multi-scale channel attention mechanism; spatial transformation; convolutional neural network

1. Introduction

Water conservancy engineering is a fundamental infrastructure closely related to human production and livelihood. In its daily management and continuous operation, it has accumulated a wealth of data. Achieving the scientific management of data related to water conservancy engineering is of paramount significance for driving the digital transformation of the water conservancy sector. A knowledge graph [1,2] is a semantic network that reveals entities, concepts and their relationships existing in the real world. It is composed of a large number of triplets (head entity, relation, tail entity). The emergence of knowledge graphs has revolutionized traditional knowledge storage methods, providing a better means to organize, manage and comprehend massive amounts of information in the field of hydraulic engineering. Currently, deep learning models find widespread application in the field of engineering [3,4]. With the continuous advancement of water information technology, the application of knowledge graph technology in the water conservancy field is gradually expanding and deepening. Yan et al. [5] constructed a water affairs knowledge graph, achieving integrated management of water-related information. Feng et al. [6] built a knowledge graph for water conservancy object data. Wang et al. [7] utilized a knowledge graph of water conservancy engineering to implement intelligent emergency plan generation. These domain-specific knowledge graphs integrated with the water conservancy industry are essential for consolidating various water resource data and promoting the intelligent development of the industry.

However, due to the multi-source heterogeneity of data in knowledge graphs [8] and the complexity of knowledge extraction processes [9–11], knowledge graphs suffer from the issue of incomplete information, severely impacting their effectiveness in practical applications [12]. Traditionally, rule-based methods have been employed to address data incompleteness in water conservancy knowledge graphs [6]. However, these methods rely on a large number of manually extracted rules, which come with the drawbacks of high cost and low efficiency. They are unable to effectively complete and optimize large-scale knowledge graphs. Among the various knowledge graph completion methods, knowledge graph embedding (KGE) models [13,14] have gained widespread attention in recent years. KGE models, by learning low-dimensional representations of knowledge graphs, make the graph computable, more easily integrable with deep learning models and better equipped to support knowledge graph completion tasks [15].

In existing research on knowledge graph embeddings, translation models often employ shallow structures and model the relationships between entities in a simple and intuitive manner. For example, TransE [16], TransH [17] and TransD [18] consider relations as translations between head and tail entities. RotatE [19] defines each relation as a rotation in the complex vector space from the head entity to the tail entity. These models can effectively learn explicit knowledge in knowledge graphs and exhibit high computational efficiency. However, due to the limitations of shallow and linear structures, these models struggle to extract latent knowledge between entity relationships and perform poorly in handling complex relationships. In recent years, owing to the powerful feature extraction capabilities of convolutional neural networks (CNNs), KGE models based on CNNs have gained increasing attention from researchers. ConvE [20] was the first model to introduce a 2D convolutional neural

network into knowledge representation tasks. However, ConvE constructs the input matrix for 2D convolution in a simple stacking manner (Figure 1(a)), limiting the interaction between entities and relationships. As shown in Figure 1(b), interaction information can only be captured when the convolutional kernel is positioned at the concatenation of entity and relationship embeddings. Subsequently, ConvR [21] reshapes relationship embeddings into convolutional kernels to achieve complete interaction between entities and relationships. InteractE [22] proposes a novel reshaping method to maximize heterogeneous interaction between entities and relationships. Improving the interaction mode enhances the expressive power of the model, so further exploration of more effective ways of interaction between entity and relationship embeddings is necessary. Furthermore, due to the non-linear operation of multiple layers of convolution, models may lose explicit knowledge during the convolution process [23]. And since convolution operates within local neighborhoods, CNNs fail to capture global semantic information. Increasing the size of convolutional kernels and the number of convolutional layers results in the loss of local detailed features and reduces efficiency. Therefore, addressing how to further enhance the capture of both local and global features is a problem that needs to be addressed.

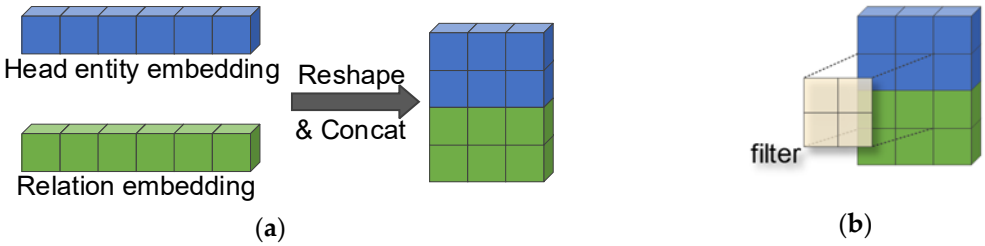


Figure 1. Partial operations of the ConvE Model. (a) Interaction mechanism between embeddings in ConvE; (b) Convolutional operation in ConvE.

Based on the aforementioned issues, this paper introduces a knowledge graph embedding model called STAM, which couples spatial transformation and attention mechanism. STAM achieves deep interaction between vectors through projection transformation and element-wise fusion. During this interaction, the model refrains from splitting embeddings into other spatial dimensions, thus preserving more linear features within the embedding vectors. Furthermore, inspired by the InteractE model, STAM reorders the internal elements of vectors before element-wise fusion, further enhancing the interaction between entity embeddings and relation embeddings. To learn the deep connections between entities and relations, STAM introduces two-dimensional convolution to process interaction embeddings. Additionally, a novel multi-scale channel attention mechanism is employed to augment the model's focus on both global information and fine-grained features, ultimately yielding feature embeddings with richer information. The contributions of the paper are as follows:

- Proposing a novel approach for the interaction between entity embedding and relationship embedding by element-wise fusion of entities and relationships on two separate hyperplanes to achieve comprehensive fusion while preserving the shallow information between entities and relationships.
- Constructing a multi-scale channel attention mechanism to fuse global deep features and local low-level features in the feature map enhances the attention of the model to key features.
- The STAM model is applied to a hydraulic engineering dataset, and experimental results

demonstrate that it outperforms representative baseline methods significantly in link prediction tasks.

2. Related work

2.1. Translation models

Inspired by the phenomenon of translation invariance in word embedding space, translation models utilize relations to explicitly represent the hidden semantic correlations between entity vectors. TransE, the first proposed translation model, has had a profound impact on the research of knowledge graph embeddings. The core idea of TransE is to consider relations as translations from the head entity to the tail entity, which is a simple and efficient approach. However, it struggles to handle complex relationships such as 1-N, N-1 and N-N. To address this problem, TransH defines separate hyperplanes for each relation and projects entities onto the hyperplane corresponding to the relation. This allows entities to have different representations under different relations, thereby enhancing the ability of the model to represent complex relationships. Unlike TransH, TransR [24] embeds entities and relations into different spaces and models triplets in the space corresponding to the relation. This enables the model to distinguish multi-faceted information contained in entities. Inspired by Euler's formula, RotateE adopts a rotation operation in a low-dimensional space to represent different types of relations by rotating the relation vector by a certain angle. This approach better handles symmetric and antisymmetric relationships in knowledge graphs.

2.2. Neural network-based methods

Translation models have strong interpretability and high computational efficiency. However, they struggle to capture deep associations between entity embeddings and relation embeddings, making them less suitable for large-scale knowledge graphs with complex structures. In recent years, with the rapid development of neural networks, a series of knowledge graph embedding models combined with deep learning techniques have been proposed [25]. ConvE utilizes a multi-layer CNN to extract local interaction features between head entity embeddings and relation embeddings. Leveraging the characteristics of convolutional networks, this convolution-based approach can extract rich semantic features and achieve high parameter utilization, significantly improving model performance. CNN-based knowledge graph embedding methods have become a current research focus. The ConvE model treats the head entity and relation as inputs to the CNN, ignoring the global features of triplets. To utilize the global features of triplets, ConvKB [26] stacks the embeddings of head entity, relation and tail entity into a three-column matrix, which is then fed into the CNN. This captures the global features and transitional features between entities and relations, achieving good results in link prediction tasks [27,28]. Addressing the limitation of ConvE and ConvKB in only extracting interaction information at the concatenation point of entities and relations, InteractE [22] proposes three main viewpoints: feature permutation, checkered reshaping and circular convolution. These maximize the heterogeneous interaction between entities and relations, enhancing the expressive power of the model and achieving more competitive results in knowledge graph completion tasks. Similarly, addressing the issue of insufficient interaction between entities and relations, ConvR [21] does not solely use static convolution for feature extraction. Instead, it reshapes the relation into a convolutional kernel and performs convolution on the reshaped entity as a 2D matrix, ensuring that the resulting feature maps

contain rich interaction features, achieving complete interaction between entities and relations. HypER [29] further enhances the model's interpretability by employing relation-specific convolutional filters generated by a hypernetwork for head entity convolution. The multi-view feature augmented neural network (MFAE) [30] combines multi-view spatial transformations with feature-enhanced convolutional networks to obtain more feature information corresponding to entities and relations, thus improving effectiveness of the model.

Translation models are capable of capturing shallow semantic information between entities and relations, but they lack the expressive power of neural network models. CNNs can capture latent features between entities and relations, but often overlook shallow information. Therefore, this paper proposes the STAM model, which utilizes spatial transformation to achieve deep interactions between entity embeddings and relation embeddings while preserving shallow information between entities and relations. Additionally, it combines CNNs with a multi-scale channel attention mechanism to fully extract latent information between entities and relations, thereby strengthening the model's expressive power.

3. Methodology

3.1. Notations

Representing the knowledge graph as $G = (E, R, T)$, where $E = \{e_1, e_2, \cdots, e_{|E|}\}$ represents the set of entities in the knowledge graph, $R = \{r_1, r_2, \cdots, r_{|R|}\}$ represents the set of relationships in the knowledge graph and $T = \{(e_h, r, e_t) \epsilon E \times R \times E\}$ represents the set of triples in the knowledge graph. In each triple, the head entity, relationship and tail entity are represented as e_h, r and e_t, respectively. Additionally, the knowledge graph embedding models always project the triples (e_h, r, e_t) onto a lower-dimensional space to obtain vector representations for the triples. In this paper, we use $k_h, k_r, k_t \in \mathbb{R}^k$ to represent the vector representations of the head entity, relationship and tail entity in the low-dimensional space, where k denotes the dimensionality of the low-dimensional space.

To evaluate the performance of the model, we adopt the link prediction task. The link prediction task involves predicting the missing entity given the head entity and relationship or predicting the missing head entity given the relationship and tail entity. It can be represented as $(e_h, r, ?)$ or $(?, r, e_t)$. We define a scoring function $\varphi(e_h, r_i, e_t) \in \mathbb{R}$ to calculate the score of a triple, where a higher score indicates a higher probability of the triple being valid.

3.2. STAM

The overall framework of STAM, as shown in Figure 2, consists of three main components: spatial transformation, multi-scale channel attention and information fusion. The model comprises two feed-forward paths, Path 1 and Path 2. In Path 1, the entities and relations go through projection, element rearrangement and element-wise interaction, resulting in a set of interaction vectors S_p. In Path 2, the entities and relations are directly subjected to element rearrangement and element-wise fusion, generating another set of interaction vectors S_r. The model applies identical operations to both paths, including convolution, channel weight adjustment, important feature extraction and feature fusion. Finally, the information from both paths is merged to obtain the final feature embedding V.

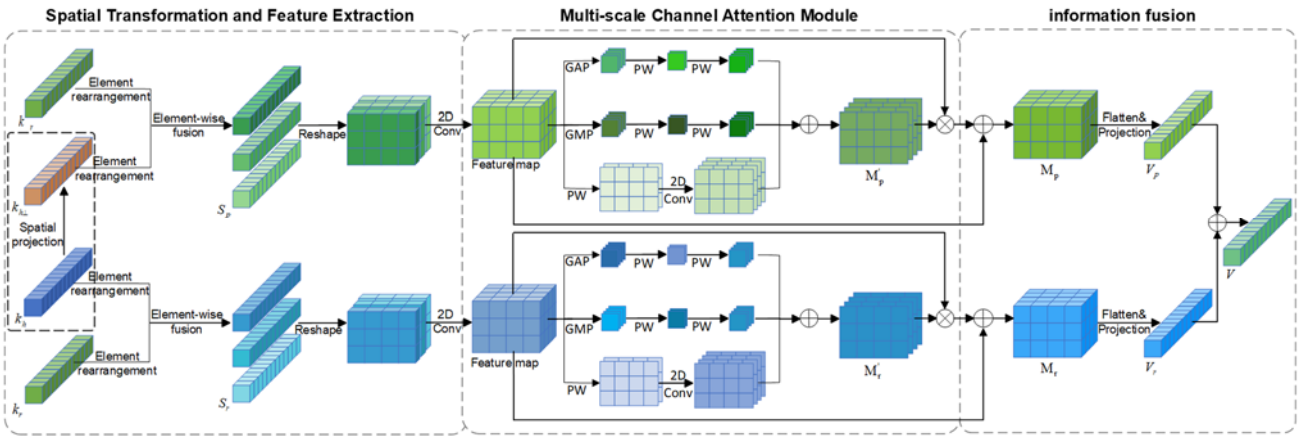


Figure 2. The architecture of STAM. Conv represents the convolution operation. GMP denotes global maximum pooling. GAP represents global average pooling. PW refers to pointwise convolution.

3.2.1. Spatial transformation

The degree of interaction between entity embeddings and relation embeddings significantly influences the potential semantic extraction capability of knowledge graph embedding models. Therefore, we achieve deep interactions between entities and relations through projection and element-wise fusion operations, resulting in two sets of interaction vectors that reside on different hyperplanes and provide complementary information. The specific interaction process is illustrated in Figure 3. Firstly, the entity embedding k_h is mapped to the hyperplane α specific to the relation embedding k_r (Figure 3(a)), obtaining the projected vector $k_{h\perp}$. Then, we fuse the element-rearranged projected vector $k_{h\perp}^i$ with the relation embedding k_r^i element-wise (Figure 3(b)), obtaining the interaction information v_p^i that emphasizes relations. By directly element-wise fusing the entity embedding k_h^i and the relation embedding k_r^i (Figure 3(b)), the direct interaction information v_r^i between entities and relations is obtained.

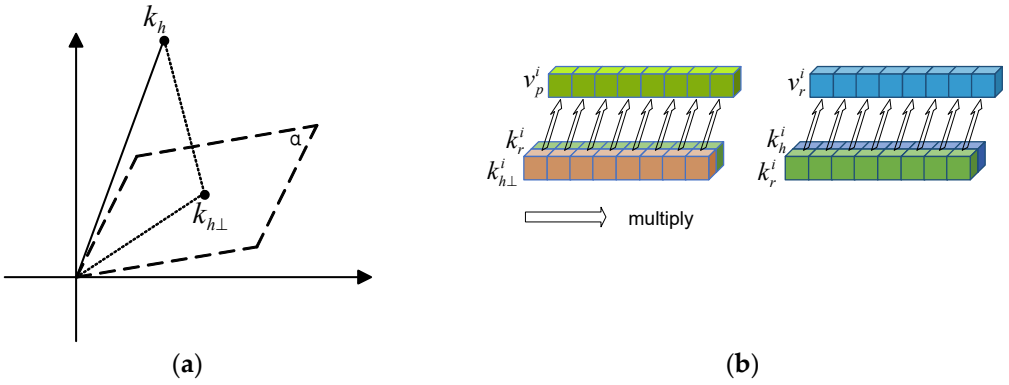
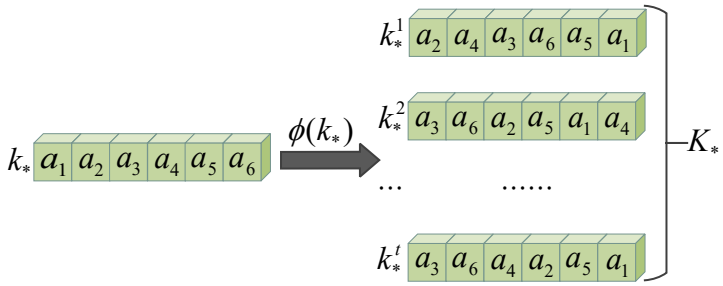


Figure 3. The interaction between entities and relations. (a) Spatial projection; (b) Element-wise fusion.

Additionally, to preserve the integrity of entity embeddings and relation embeddings and retain

more linear features and shallow information, we do not split the embeddings into other dimensions in the space. And to further capture richer interaction information between entities and relations, we perform t rounds of rearrangement on the internal elements of the entity embedding k_h, relation embedding k_r and the projected vector $k_{h\perp}$. The specific operation is illustrated in Figure 4, resulting in a new set of entity embeddings K_h, relation embeddings K_r and projected vectors $K_{h\perp}$.



*: General representation, representing h, r and $h\perp$.
$\phi(k_*)$: Rearrange the elements of vector k_* randomly t times.

Figure 4. Rearrangement of elements in a vector.

The specific process is as follows:

$$K_h = \left[k_h^1, k_h^2, \cdots, k_h^t\right] = \phi(k_h) \quad K_r = [k_r^1, k_r^2, \cdots, k_r^t] = \phi(k_r), \tag{1}$$

$$k_{h\perp} = (k_h - W_r^T k_h W_r) \quad K_{h\perp} = \left[k_{h\perp}^1, k_{h\perp}^2, \cdots, k_{h\perp}^t\right] = \phi(k_{h\perp}), \tag{2}$$

$$S_p = \left[v_p^1, v_p^2, \cdots, v_p^t\right] = K_{h\perp} \circ K_r, \tag{3}$$

$$S_r = [v_r^1, v_r^2, \cdots, v_r^t] = K_h \circ K_r, \tag{4}$$

where $\phi(\cdot)$ represents t random element permutations applied to the vectors. $k_h^i, k_r^i, k_{h\perp}^i \epsilon \mathbb{R}^k$ denotes the entity embeddings, relation embeddings and projection vectors after internal element rearrangements. $K_h \epsilon \mathbb{R}^{t\times k}$ represents a new set of entity embeddings. $K_r \epsilon \mathbb{R}^{t\times k}$ represents a new set of relation embeddings. $K_{h\perp} \epsilon \mathbb{R}^{t\times k}$ represents a new set of projection vectors. $\circ$ denotes the Hadamard product. $W_r \epsilon \mathbb{R}^k$ is a trainable parameter denoting the normalized normal vector of hyperplane α corresponding to relation k_r. In our model, each relation corresponds to a unique hyperplane, and the normal vector of the hyperplane is initialized by the model and adjusted and optimized during the training process.

The v_p^i and v_r^i in S_p and S_r are reshaped into two-dimensional data resembling images, denoted as $re(S_p)$ and $re(S_r)$, respectively. Then, a two-dimensional convolution operation is performed on each of them to obtain feature maps. The process is as follows:

$$S_p \in \mathbb{R}^{t\times k} \rightarrow re(S_p) \in \mathbb{R}^{t\times k_w \times K_h} \quad S_r \in \mathbb{R}^{t\times k} \rightarrow re(S_r) \in \mathbb{R}^{t\times k_w \times K_h}, \tag{5}$$

$$M_p = re(S_p) * \omega_p, \tag{6}$$

$$M_r = re(S_r) * \omega_r, \tag{7}$$

where $k_h \times k_w = k$, $*$ denotes the convolution operation, $\omega_p, \omega_r \epsilon \mathbb{R}^{c\times p\times q}$ are the convolution kernels,

c represents the number of filters, p and q denote the height and width of the filters and M_p, $M_r \epsilon \mathbb{R}^{c\times(k_h-p+1)\times(k_w-q+1)}$ are the feature maps.

3.2.2. Multi-scale channel attention

The paper introduces a multi-scale channel attention mechanism, which computes attention weights for each channel of feature maps at multiple scales. This mechanism enables the model to capture richer contextual information, fine-grained features and multi-scale information, thereby enhancing the representation ability of the model for entities and relations. The overall framework of the multi-scale channel attention mechanism is shown in Figure 5.

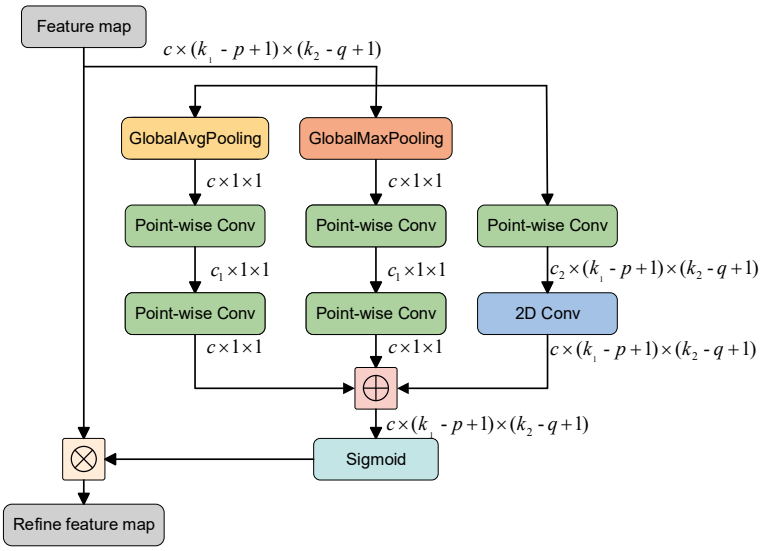


Figure 5. The overall framework of the multi-scale channel attention mechanism.

Firstly, global max pooling and global average pooling are used to aggregate global information, resulting in the global channel context, G_p and G_r, for the two pathways. The specific operations are as follows:

$$G_p' = GMP(M_p) * \omega_1 * \omega_2, \tag{8}$$

$$G_p'' = GAP(M_p) * \omega_1 * \omega_2, \tag{9}$$

$$G_p = G_p' + G_p'', \tag{10}$$

$$G_r' = GMP(M_r) * \omega_1 * \omega_2, \tag{11}$$

$$G_r'' = GAP(M_r) * \omega_1 * \omega_2, \tag{12}$$

$$G_r = G_r' + G_r'', \tag{13}$$

where $GMP(\cdot)$ represents global max pooling. $GAP(\cdot)$ represents global average pooling. $\omega_1 \in \mathbb{R}^{c_1\times1\times1}$ is the convolution kernel for the first convolution, with c_1 indicating the number of filters

and 1×1 indicating the height and width of the filters. $\omega_2 \in \mathbb{R}^{c\times1\times1}$ is the convolution kernel for the second convolution, with c indicating the number of filters and 1×1 indicating the height and width of the filters, where $c > c_1$.

Next, we employ pointwise convolution and regular convolution to aggregate local information, obtaining local channel contexts L_p and L_r on the two paths.

$$L_p = M_p * \omega_3 * \omega_4, \tag{14}$$

$$L_r = M_r * \omega_3 * \omega_4, \tag{15}$$

where $\omega_3 \in \mathbb{R}^{c_2\times1\times1}$ is the kernel of the first convolution with c_2 representing the number of filters and 1×1 representing the height and width of the filters. $\omega_4 \in \mathbb{R}^{c\times3\times3}$ is the kernel of the second convolution, where c represents the number of filters, 3×3 represents the height and width of the filters and $c > c_2$. It is worth noting that in both convolution operations, zero-padding is used to ensure that L_p,$L_r \in \mathbb{R}^{c\times(k_h-p+1)\times(k_w-q+1)}$, and the input features have the same shape, preserving fine-grained information in the shallow features.

Finally, the refined features $M_r', M_p' \in \mathbb{R}^{c\times(k_h-p+1)\times(k_w-q+1)}$, which incorporate both local and global contextual information, are obtained by adding L_p and L_r to the global contexts G_p and G_r within the attention module:

$$M_p' = M_p \circ \sigma\big(L_p \oplus G_p\big), \tag{16}$$

$$M_r' = M_r \circ \sigma(L_r \oplus G_r), \tag{17}$$

where $\sigma(\cdot)$ represents the sigmoid activation function and $\oplus$ denotes the broadcasting addition.

3.2.3. Information fusion

As mentioned earlier, 2D convolution can extract complex latent information between entity relationships, but deep-level operations ignore surface explicit knowledge. To obtain richer semantic information, we combine the refined features M_p' and M_r' with the surface explicit features M_p and M_r. Then, we flatten and project the resulting features into a k-dimensional space, obtaining V_p and V_r.

$$V_p = f\big(vec(M_p' + M_p)W + b\big), \tag{18}$$

$$V_r = f(vec(M_r' + M_r)W + b), \tag{19}$$

where $V_p, V_r \in \mathbb{R}^k$. $vec(\cdot)$ denotes the vectorization operation. $f(\cdot)$ represents the ReLU activation function. $W \in \mathbb{R}^{\boldsymbol{k}\times n}$ and $b \in \mathbb{R}^{\boldsymbol{k}}$ are parameters of the fully connected layer, with $n = c \times (k_h - p + 1) \times (k_w - q + 1)$.

The final feature embedding V is obtained by element-wise addition of the results V_p and V_r from the two pathways.

$$V = V_p + V_r, \tag{20}$$

where $V \in \mathbb{R}^k$.

3.2.4. Scoring function

By taking the inner product between the feature embedding V and the tail entity vector k_t, we calculate the score of the input triplet (e_h, r, e_t). The scoring function of STAM is defined as follows:

$$\varphi(e_h, r, e_t) = \sigma(V \cdot k_t + b) \in (0,1), \tag{21}$$

where $\sigma(\cdot)$ represents the sigmoid function. $\cdot$ denotes matrix multiplication. $b \in \mathbb{R}$ represents the bias term. During training, the standard binary cross-entropy is used to compute the loss.

4. Experiments

4.1. Datasets

In this study, we validate the effectiveness of the model using WN18RR [20], FB15K-237 [31], Kinship [32] and a hydraulic engineering dataset. WN18RR and FB15K-237 are the most widely used general-domain datasets in recent years for knowledge graph completion tasks. WN18RR is derived from the WN18 [33] dataset, while FB15K-237 is derived from the FB15K [16] dataset. Due to the presence of numerous inverse relations in the WN18 and FB15K datasets, models can achieve very advanced results through simple reverse rule application, making it difficult to effectively distinguish the superiority of different models. To ensure the accuracy of experimental results, Dettmers et al. [20] removed triplets with inverse relations from the WN18 and FB15K datasets, resulting in the WN18RR and FB15K-237 datasets. We also utilized two domain-specific datasets, namely the Kinship dataset from the social domain and a hydraulic engineering dataset, to investigate the model's generalization ability on specific domain datasets.

WN18RR consists of 11 relations and 40,943 entities. FB15K-237 has 237 relations and 14,541 entities. Kinship is a small dataset describing kinship relationships, consisting of 25 relations and 104 entities. The hydraulic engineering dataset (HEDS) includes hydraulic engineering cases, fundamental knowledge of hydraulic engineering, hydraulic geographical information, totaling 6088 entities and 509 relations. To conduct model training, evaluation and tuning, the article divides the hydraulic engineering dataset into a training set (70%), a test set (20%) and a validation set (10%). The statistical information of the datasets is presented in Table 1.

Table 1. The statistical information of the datasets.

Dataset	Entities	Relations	Triplets		
			Train	Valid	Test
WN18RR	40,943	11	86,835	3034	3134
FB15K-237	14,541	237	272,115	17,535	20,446
Kinship	104	25	8544	1068	1074
HEDS	6088	509	15,662	4475	2237

4.2. Evaluation metrics

In the experiments, the performance of the model is evaluated using mean reciprocal rank (MRR) and Hits@n metrics. For a given test triplet (e_h, r, e_t), candidate triplets are generated by replacing the

tail entity e_t with all other entities $e \in E$ in the knowledge graph. The test triplet (e_h, r, e_t) and candidate triplets are then scored together, and the scores are ranked in descending order. Hits@n represents the proportion of test triplets whose score ranks are less than or equal to n. The calculation formulas are as follows:

$$Hits@n = \frac{1}{|M|}\sum_{i=0}^{|M|} I(rank_i \leq n), \tag{22}$$

where M represents the test set. $|M|$ represents the number of triplets in the test set. $rank_i$ represents the ranking of the i-th test triplet in the link prediction task. $I(\cdot)$ denotes the indicator function, which takes the value of 1 when the condition is true and 0 otherwise.

MRR represents the average reciprocal rank of all test triplets. The calculation formula is as follows:

$$MRR = \frac{1}{|M|}\sum_{i=1}^{|M|} \frac{1}{rank_i}, \tag{23}$$

The symbols have the same meanings as Hits@n. When a knowledge graph embedding model achieves high Hits@n and MRR values, it indicates better performance. It is worth noting that MRR not only considers whether the correct triplets appear in the top-k rankings but also takes into account their positions in the ranking list, providing a more comprehensive evaluation of the performance of knowledge graph embedding models. Therefore, this paper focuses more on the MRR metric.

4.3. Experimental settings

In this paper, grid search was used for hyperparameter tuning. The Adam optimizer was employed for training, with 1000 iterations on the public datasets FB15K-237, WN18RR and Kinship, and 500 iterations on the hydraulic engineering dataset. The hyperparameter ranges for grid search were as follows: learning rates were {0.0005, 0.001, 0.0008}, dropout rates for the input layer, feature maps and hidden layers were {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8}, the number of convolutional kernels were {32, 64, 96, 128, 256} and the kernel sizes were {3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11}.

The entity embedding dimension and relation embedding dimension were set to 400 for all datasets. The hyperparameter settings for STAM on FB15K-237, WN18RR, Kinship and the hydraulic engineering dataset are presented in Table 2.

Table 2. The hyperparameter settings on different datasets.

Dataset	Learning rate	Batch size	Dropout			Kernel size	Embedding size	Kernel number
			Input	Feature map	Hidden layer			
FB15K-237	0.001	128	0.3	0.6	0.3	9 × 9	400	96
WN18RR	0.001	128	0.2	0.6	0.2	9 × 9	400	96
Kinship	0.0005	128	0.7	0.1	0.8	9 × 9	400	96
HEDS	0.001	128	0.1	0.2	0.4	7 × 7	400	32

4.4. Results and discussion

The paper first evaluates the link prediction capability of STAM on WN18RR and FB15k-237 datasets, comparing it against several baseline models: translation models (TransE, TransR, RotatE), neural network-based models (ConvE, HypER, ConvR, interactE, MFAE).

The final experimental results are presented in Table 3, where the best results are shown in bold and the second-best results are underlined. It can be observed that on the FB15k-237 dataset, STAM outperforms all other baseline models on the four evaluation metrics. Compared to ConvE, STAM achieves notable improvements of 10.5% in MRR and 12.2% in Hits@1. In comparison with InteractE and MFAE, STAM exhibits stronger competitiveness on the four evaluation metrics, indicating its ability to capture more diverse and effective information. On the WN18RR dataset, STAM outperforms the best-performing neural network-based baseline model MFAE on all metrics except Hits@1. This indicates that introducing spatial transformation and attention mechanisms in CNNs can effectively capture the complex relationships between entities, leading to significant improvements in link prediction experiments.

Overall, neural network-based models demonstrate better ability to capture the semantic relationships between entities and relations, making them more advantageous in handling large-scale knowledge graphs compared to translation models. However, on the WN18RR dataset, RotatE outperforms neural network-based models in terms of MRR, Hits@10 and Hits@3. We attribute this to the relatively sparse relations in WN18RR, which align well with RotatE's proposed complex space rotation operation. On the other hand, due to the more complex relations in FB15K-237, RotatE performs poorly on this dataset, with substantially lower scores than STAM on the four evaluation metrics. In conclusion, STAM achieves favorable performance on both WN18RR and FB15k-237 datasets, demonstrating its strong generalization capability and its ability to effectively model complex relationships.

Table 3. Link prediction results of models on WN18RR and FB15k-237.

Models	WN18RR				FB15K-237			
	MRR	**Hits@10**	**Hits@3**	**Hits@1**	**MRR**	**Hits@10**	**Hits@3**	**Hits@1**
TransE	0.192	0.435	0.367	0.21	0.288	0.478	0.325	0.192
TransR	0.401	0.465	0.389	0.345	0.263	0.428	0.267	0.168
RotatE	**0.474**	**0.571**	**0.490**	0.425	0.318	0.525	0.358	0.216
ConvE	0.430	0.520	0.440	0.400	0.325	0.501	0.356	0.237
HypER	0.465	0.522	0.477	0.436	0.341	0.520	0.376	0.252
ConvR	0.467	0.524	0.480	**0.437**	0.350	0.528	0.385	0.261
interactE	0.463	0.528	0.481	0.430	0.354	0.535	0.390	0.263
MFAE	0.467	0.530	0.482	<u>0.437</u>	<u>0.355</u>	<u>0.540</u>	<u>0.390</u>	<u>0.263</u>
STAM	<u>0.468</u>	<u>0.535</u>	<u>0.484</u>	0.435	**0.359**	**0.541**	**0.394**	**0.266**

The evaluation of STAM on the small-scale Kinship dataset was also conducted, comparing it with baseline models TransE, RotatE, ConvE, ConvR, InteractE and MFAE. The experimental results are presented in Table 4. It can be observed that STAM outperforms all baseline models on the four evaluation metrics. Additionally, due to the limited number of relations in the Kinship dataset (only 25 relations), RotatE performs well and achieves results second only to STAM in terms of MRR and

Hits@1. Notably, the previously successful models, interactE and MFAE, which exhibited excellent performance on the WN18RR and FB15K-237 datasets, show poor performance on the Kinship dataset, scoring even lower than the proposed ConvE model based on 2D convolutional neural networks. STAM outperformed InteractE and MFAE in MRR by 14.2% and 24.6%, respectively, and exceeded the best-performing convolutional model, ConvR, by 6.6% in terms of MRR. These results demonstrate that STAM's performance on the Kinship dataset is superior to the currently advanced convolutional neural network models. It can extract meaningful features from limited information and achieve effective modeling of specific domain datasets, showcasing strong generalization capability.

Table 4. Link prediction results of models on Kinship.

Models	Kinship			
	MRR	**Hits@10**	**Hits@3**	**Hits@1**
TransE	0.309	0.841	0.643	0.090
RotatE	<u>0.843</u>	0.978	0.919	<u>0.760</u>
ConvE	0.830	<u>0.980</u>	<u>0.920</u>	0.740
ConvR	0.832	0.964	0.912	0.750
interactE	0.777	0.959	0.870	0.664
MFAE	0.712	0.893	0.797	0.606
STAM	**0.887**	**0.986**	**0.949**	**0.821**

To further validate the effectiveness of the STAM model in completing the knowledge graph in hydraulic engineering, this paper evaluated the STAM model on a hydraulic engineering dataset. The experimental results are shown in Table 5. It can be observed that convolutional neural network-based models achieve better results compared to traditional translation-based models, indicating that CNNs can effectively capture rich nonlinear features between entities and relations, leading to better fitting performance on complex datasets with a large number of entity and relation objects. Moreover, in the link prediction task on the water conservancy engineering knowledge graph, the proposed STAM model outperforms the baseline models in all evaluation metrics. Compared to the best-performing convolutional models, InteractE and MFAE, STAM achieves a 3.1% and 2% improvement in MRR, respectively. This demonstrates that our model captures more effective explicit features and latent information between entities and relations during feature extraction, making it suitable for completing tasks on water conservancy engineering knowledge graphs with complex relationships.

Table 5. Link prediction results of the model on the hydraulic engineering dataset.

Models	Hydraulic engineering dataset			
	MRR	**Hits@10**	**Hits@3**	**Hits@1**
TransE	0.085	0.179	0.091	0.040
RotatE	0.091	0.177	0.098	0.040
ConvE	0.349	0.661	0.406	0.207
ConvR	0.336	0.640	0.394	0.197
interactE	0.354	0.666	0.416	0.210
MFAE	<u>0.358</u>	<u>0.672</u>	<u>0.421</u>	<u>0.214</u>
STAM	**0.365**	**0.680**	**0.429**	**0.220**

4.5. Ablation study

To validate the effectiveness of each module in the STAM model, ablative experiments were conducted on a hydraulic engineering dataset. The experimental results are presented in Table 6, with the best results highlighted in bold. STAM-Att denotes the removal of the multi-scale channel attention module from the model. STAM-ST indicates the exclusion of the spatial transformation module from the model. STAM-Sp represents the removal of the interaction vector S_p from the spatial transformation module. STAM-Sr represents the removal of the interaction vector S_r from the spatial transformation module.

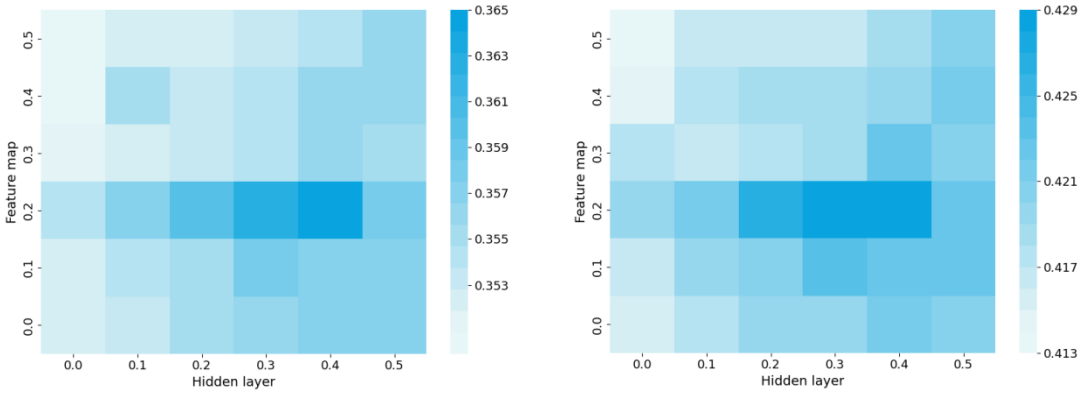
The results demonstrate that the STAM model achieves optimal performance across all metrics. When the multi-scale channel attention is removed, the overall performance of the model declines, indicating that attention mechanisms effectively integrate local detailed features and global semantic features, thereby enhancing the model's expressive power. In ablative experiments on the spatial transformation module, direct element-wise addition is used to fuse entity vectors and relation vectors. However, as evident from Table 6, the model fails to achieve the best performance, validating the proposition that the spatial transformation module proposed in this study more effectively integrates entity embeddings and relation embeddings. Comparing STAM with STAM-Sp and STAM-Sr, it is observed that removing the interaction vectors S_p and S_r from the module leads to a decrease in model performance. This further confirms the effectiveness of the spatial transformation module and verifies the necessity of the two sets of interaction vectors in the spatial transformation module.

Table 6. Results of ablation study.

Models	Hydraulic engineering dataset			
	MRR	**Hits@10**	**Hits@3**	**Hits@1**
STAM-Sr	0.359	0.673	0.42	0.215
STAM-Sp	0.358	0.672	0.42	0.214
STAM-Att	0.362	0.678	0.426	0.217
STAM-ST	0.333	0.643	0.388	0.194
STAM	**0.365**	**0.680**	**0.429**	**0.220**

4.6. Parameter sensitivity analysis

Dropout rate settings: To analyze the impact of dropout rates on the performance of STAM, we set the dropout rate range for feature maps and hidden layers as [0.0, 0.5] and conducts experiments on the hydraulic engineering dataset. Figure 6 shows the performance of STAM in terms of MRR and Hits@3 under different dropout rates. The results indicate that when the model applies dropout operations to both the hidden layers and feature maps, it achieves better performance, suggesting that setting appropriate dropout rates for feature maps and hidden layers can effectively enhance the model's expressive power. It is worth noting that the model achieves the best results on the hydraulic engineering dataset when the dropout rates for feature maps and hidden layers are set to 0.2 and 0.4, respectively.



(**a**) (**b**)

Figure 6. The impact of dropout rates on the experimental results. (a) Performance of MRR; (b) Performance of Hits@3.

Number of convolutional kernel settings: To analyze the impact of the number of convolutional kernels on the performance of the STAM model, we conducted experiments on the hydraulic engineering dataset with different numbers of convolutional kernels: 16, 32, 64, 96 and 128, while keeping other parameters unchanged. Figure 7 illustrates the performance of STAM in terms of MRR, Hits@10, Hits@3 and Hits@1 under different numbers of convolutional kernels. From the Figure 7, it can be observed that the number of convolutional kernels has a certain impact on the model's performance. When the number of kernels is too small, such as setting it to 16, the limited feature combinations restrict the model's expressive power. On the other hand, when the number of kernels is too large, such as setting it to 128, the model may overfit and fail to achieve optimal performance. According to the experimental results, the model achieves the best performance when the number of convolutional kernels is set to 32, reaching the optimal values in all metrics except Hits@1.

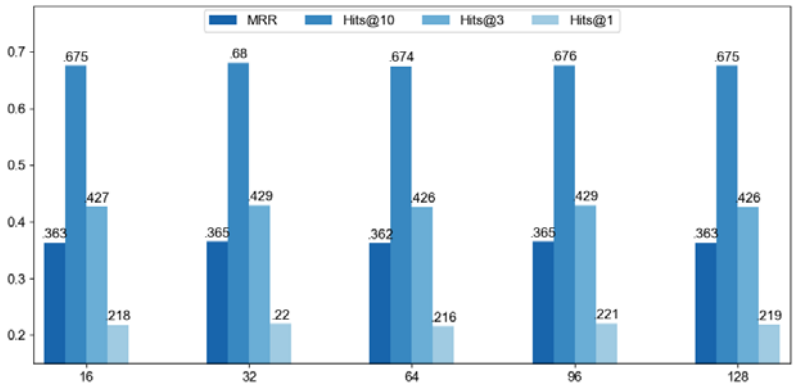


Figure 7. The influence of the number of convolutional kernels on the experimental results.

Convolutional kernel size settings: To analyze the impact of convolutional kernel size on the performance of the STAM model, we conducted experiments with different convolutional kernel sizes of 3×3, 5×5, 7×7, 9×9 and 11×11 on the hydraulic engineering dataset. Figure 8 illustrates

the performance of STAM in terms of MRR, Hits@10, Hits@3 and Hits@1 under different convolutional kernel sizes. The results indicate that the model achieves optimal performance when the convolutional kernel size is 7 × 7. Using smaller convolutional kernel sizes, such as 3 × 3 and 5 × 5, limits the number of interactions. On the other hand, larger convolutional kernel sizes, such as 9 ×9 and 11 × 11, lead to overfitting and fail to achieve the model's best performance.

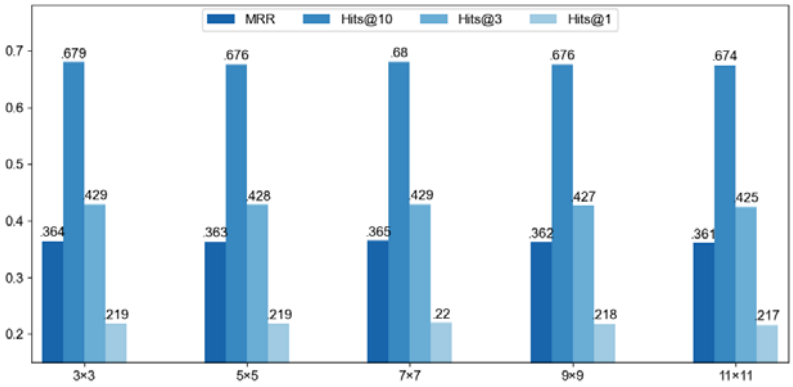


Figure 8. The influence of convolutional kernel sizes on experimental results.

5. Conclusions

This paper proposes a novel knowledge graph embedding model called STAM, which is based on spatial transformation and attention mechanism. STAM leverages spatial transformation for interactive fusion of entity relations, effectively preserving the explicit surface features between entity relations. Additionally, it incorporates a multi-scale channel attention mechanism, allowing the model to simultaneously focus on local details and global features, capturing complex relationships among entities in the knowledge graph to a great extent. We conducted experiments comparing STAM with classical knowledge graph embedding models on public datasets such as WN18RR, FB15K-237 and Kinship, and achieved promising results. Furthermore, STAM outperformed other models on the hydraulic engineering dataset, demonstrating its strong generalization ability. In this study, the model emphasizes the extraction of entity relation features within triplets. In future work, we plan to incorporate more external knowledge into the model, such as descriptive information and attribute information of hydraulic engineering objects, to further enhance the performance of the model in the task of hydraulic engineering knowledge graph completion.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Yan, C. Wang, W. Cheng, M. Gao, A. Zhou, A retrospective of knowledge graphs, *Front. Comput. Sci.*, **12** (2018), 55–74. https://doi.org/10.1007/s11704-016-5228-9
2. S. Ji, S. Pan, E. Cambria, P. Marttinen, P. S. Yu, A survey on knowledge graphs: Representation, acquisition, and applications, *IEEE Trans. Neural Networks Learn. Syst.*, **33** (2022), 494–514. https://doi.org/10.1109/tnnls.2021.3070843
3. N. Sitapure, J. S. I. Kwon, Design, exploring the potential of time-series transformers for process modeling and control in chemical systems: An inevitable paradigm shift?, *Chem. Eng. Res. Des.*, **194** (2023), 461–477. https://doi.org/10.1016/j.cherd.2023.04.028
4. N. Sitapure, J. S. I. Kwon, Crystalgpt: Enhancing system-to-system transferability in crystallization prediction and control using time-series-transformers, *Comput. Chem. Eng.*, **177** (2023), 108339. https://doi.org/10.1016/j.compchemeng.2023.108339
5. J. Yan, T. Lv, Y. Yu, Construction and recommendation of a water affair knowledge graph, *Sustainability*, **10** (2018), 3429. https://doi.org/10.3390/su10103429
6. J. Feng, X. Xu, J. Lu, Construction and application of water conservancy information knowledge graph, *Comput. Modernization*, **9** (2019), 35–40. https://doi.org/10.3969/j.issn.1006-2475.2019.09.007
7. L. Wang, X. Liu, Y. Liu, H. Li, J. Liu, L. Yang, Knowledge graph-based method for intelligent generation of emergency plans for water conservancy projects, *IEEE Access*, **11** (2023), 84414–84429. https://doi.org/10.1109/access.2023.3302399
8. E. Iglesias, S. Jozashoori, M. E. Vidal, Scaling up knowledge graph creation to large and heterogeneous data sources, *J. Web Semant.*, **75** (2023), 100755. https://doi.org/10.1016/j.websem.2022.100755
9. S. Yang, S. Yoo, O. Jeong, Denert-KG: Named entity and relation extraction model using DQN, knowledge graph, and BERT, *Appl. Sci.-Basel*, **10** (2020), 6429. https://doi.org/10.3390/app10186429
10. T. Al-Moslmi, M. G. Ocana, A. L. Opdahl, C. Veres, Named entity extraction for knowledge graphs: A literature overview, *IEEE Access*, **8** (2020), 32862–32881. https://doi.org/10.1109/access.2020.2973928
11. Z. Geng, Y. Zhang, Y. Han, Joint entity and relation extraction model based on rich semantics, *Neurocomputing*, **429** (2021), 132–140. https://doi.org/10.1016/j.neucom.2020.12.037
12. T. Shen, F. Zhang, J. Cheng, A comprehensive overview of knowledge graph completion, *Knowledge-Based Syst.*, **255** (2022), 109597. https://doi.org/10.1016/j.knosys.2022.109597
13. A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, *ACM Trans. Knowl. Discovery Data*, **15** (2021). https://doi.org/10.1145/3424672
14. Y. Dai, S. Wang, N. N. Xiong, W. Guo, A survey on knowledge graph embedding: Approaches, applications and benchmarks, *Electronics*, **9** (2020), 750. https://doi.org/10.3390/electronics9050750
15. Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, Z. Duan, Knowledge graph completion: A review, *IEEE Access*, **8** (2020), 192435–192456. https://doi.org/10.1109/access.2020.3030076

16. A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, **26** (2013).
17. Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **28** (2014). https://doi.org/10.1609/aaai.v28i1.8870
18. G. Ji, S. He, L. Xu, K. Liu, J. Zhao, 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)*, (2015), 687–696. https://doi.org/10.3115/v1/P15-1067
19. Z. Sun, Z. H. Deng, J. Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, preprint, arXiv:1902.10197. https://doi.org/10.48550/arXiv.1902.10197
20. T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2D knowledge graph embeddings, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **32** (2018). https://doi.org/10.1609/aaai.v32i1.11573
21. X. Jiang, Q. Wang, B. Wang, Adaptive convolution for multi-relational learning, 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)*, (2019), 978–987. https://doi.org/10.18653/v1/N19-1103
22. S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, P. Talukdar, Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2020), 3009–3016. https://doi.org/10.1609/aaai.v34i03.5694
23. Z. Zhou, C. Wang, Y. Feng, D. Chen, JointE: Jointly utilizing 1D and 2D convolution for knowledge graph embedding, *Knowledge-Based Syst.*, **240** (2022), 108100. https://doi.org/10.1016/j.knosys.2021.108100
24. S. Jia, Y. Xiang, X. Chen, K. Wang, Triple trustworthiness measurement for knowledge graph, in *the World Wide Web Conference*, (2019), 2865–2871. https://doi.org/10.1145/3308558.3313586
25. Z. Zhang, Z. Li, H. Liu, N. N. Xiong, Multi-scale dynamic convolutional network for knowledge graph embedding, *IEEE Trans. Knowl. Data Eng.*, **34** (2020), 2335–2347. https://doi.org/10.1109/TKDE.2020.3005952
26. D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, D. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, (2017), 327–333. http://dx.doi.org/10.18653/v1/N18-2053
27. L. Thanh, L. Nam, L. Bac, Knowledge graph embedding by relational rotation and complex convolution for link prediction, *Expert Syst. Appl.*, **214** (2023). https://doi.org/10.1016/j.eswa.2022.119122
28. M. Nayyeri, G. M. Cil, S. Vahdati, F. Osborne, M. Rahman, S. Angioni, et al., Trans4E: Link prediction on scholarly knowledge graphs, *Neurocomputing*, **461** (2021), 530–542. https://doi.org/10.1016/j.neucom.2021.02.100

29. I. Balažević, C. Allen, T. M. Hospedales, Hypernetwork knowledge graph embeddings, in *Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks*, Springer, **11731** (2019), 553–565. https://doi.org/10.1007/978-3-030-30493-5_52
30. D. Jiang, R. Wang, L. Xue, J. Yang, Multiview feature augmented neural network for knowledge graph embedding, *Knowledge-Based Syst.*, **255** (2022), 109721. https://doi.org/10.1016/j.knosys.2022.109721
31. K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (2015), 1499–1509. https://doi.org/10.18653/v1/D15-1174
32. X. V. Lin, R. Socher, C. Xiong, Multi-hop knowledge graph reasoning with reward shaping, preprint, arXiv:1808.10568. https://doi.org/10.48550/arXiv.1808.10568
33. A. Bordes, X. Glorot, J. Weston, Y. Bengio, A semantic matching energy function for learning with multi-relational data: Application to word-sense disambiguation, *Mach. Learn.*, **94** (2014), 233–259. https://doi.org/10.1007/s10994-013-5363-6

AIMS Press