Knowledge-Based Systems 325 (2025) 113974

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

MFLM-GCN: Multi-relation Fusion and Latent-relation Mining Graph Convolutional Network for entity alignment

Wei Ai [a], Yulu Liu [a], Chen Wei [a], Tao Meng [a],*, Hongen Shao [b], Zhixiong He [c], Keqin Li [d]

[a] College of Computer and Mathematics, Central South University of Forestry and Technology, Changsha Hunan, 410004, China
[b] School of Future Technology, South China University of Technology, Guangdong, Guangzhou 510641, China
[c] School of Business, Central South University of Forestry and Technology, Changsha Hunan, 410004, China
[d] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Entity alignment
Graph convolutional network
Graph random walk
Knowledge graph
Multi-relation

ABSTRACT

Entity alignment (EA) is the task of identifying equivalent entities in two knowledge graphs (KGs) using a limited set of seed entities. Existing research mainly uses graph neural networks (GNNs) to aggregate entity neighborhood features for representation to achieve better entity alignment. However, most of them ignore the fusion of multiple relations between entities and the mining of latent relations, which limits the effectiveness of entity representation to some extent. Therefore, this paper proposes a novel multi-relation fusion and latent-relation mining graph convolutional network (MFLM-GCN) for entity alignment. Specifically, first, we use seed entity pairs to establish the connection between two knowledge graphs and enhance semantic consistency with the help of local isomorphism. Second, we screen potential important related entities through graph random walks and fuse multiple local and global relationships to obtain a preliminary representation of the entity. Third, we use a multi-head attention mechanism to generate multiple association graphs, prune them and construct a densely connected layer to fully explore the deep potential relationships between entities and obtain a multi-branch representation of the entity. Finally, we use linear fusion to obtain the final embedding of the entity and achieve entity alignment. In experiments on multiple real-world datasets, the MFLM-GCN method effectively improves the entity alignment performance by enhancing the entity node representation. The source code for our method is openly accessible on GitHub at the following link: https://github.com/meng-tao/MFLR-GCN.

1. Introduction

Knowledge graphs, as a structured form of knowledge storage, has received extensive attention from academia and industry [1–3]. At present, there are knowledge graph datasets of different industries, domains, and languages. For example, the multilingual knowledge graph DBpedia [4] from Wikipedia pages, the large-scale Chinese concept graph CN-Probase [5], the most important human knowledge base Cyc [6], etc. Due to the different emphases of knowledge graphs in gathering knowledge, variations exist in the descriptions of the same entity across different knowledge graphs. These differences contribute to a certain degree of complementarity among the knowledge graphs.

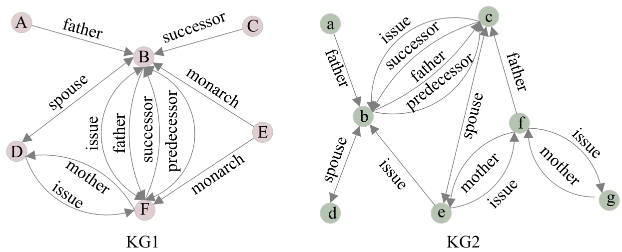
Knowledge fusion aims to consolidate entities representing identical concepts [7], integrate knowledge from diverse sources into a cohesive and succinct knowledge base, and foster interoperability among applications utilizing disparate knowledge graphs. Consequently, knowledge fusion plays a crucial role in extracting fundamental insights for downstream tasks, including link prediction [8], recommendation systems [1], and question-answering [9].

Existing research proposes that entity alignment serves as a mechanism for accomplishing knowledge fusion [10–12]. Entity alignment aims to correspond entities in different knowledge graphs so as to measure the "distance" between them in the shared embedding space and realize the structural alignment between knowledge graphs [13]. As depicted in Fig. 1, KG1 and KG2 represent knowledge graphs in different languages within the real-world context. Node "F" in KG1 and Node "b" in KG2 correspond to the identical entity "Philip", despite being expressed in unequal languages. The objective of the entity alignment task is to effectively establish a correspondence between these two nodes. Existing entity alignment methods based on representation learning are mainly divided into the following two types:

Translation-Based Representation. This type of method [14–16] mainly projects entities and relationships into a low-dimensional vector

* Corresponding author.
E-mail addresses: aiwei@hnu.edu.cn (W. Ai), liuyulu@csuft.edu.cn (Y. Liu), weichen@csuft.edu.cn (C. Wei), mengtao@hnu.edu.cn (T. Meng), ftshaohongen@mail.scut.edu.cn (H. Shao), hezhixiong@csuft.edu.cn (Z. He), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.knosys.2025.113974
Received 19 January 2025; Received in revised form 11 June 2025; Accepted 14 June 2025
Available online 28 June 2025
0950-7051/© 2025 Elsevier B.V. All rights reserved, including those for text and data mining, AI training, and similar technologies.



DESCRIPTIONS

Nodes	Full name	Nodes	Full name
A	Prince_Laurent_of_Belgium	a	艾蕾歐諾拉_（比利時公主）
B	Albert_II_of_Belgium	b	菲利普_（比利时国王）
C	Baudouin_of_Belgium	c	阿尔贝二世_（比利时）
D	Queen_Paola_of_Belgium	d	瑪蒂爾德_（比利時王后）
E	Elio_Di_Rupo	e	保拉_（比利時王后）
F	Philippe_of_Belgium	f	阿斯特里德公主_（比利时）
		g	路易莎・玛丽亚公主

Fig. 1. Examples of real-world knowledge graphs in different languages. It is obvious that there are multiple display relationships between entities, and there are rich potential relationships between entities. For example, in KG1, entities "A" and "F" are not directly related, but through entity "B", it is easy to know that there is a potential relationship between them.

space through the approximate representation of $h + r \approx t$ to learn their vector space representation. Among them, "h" is the head entity, "t" is the tail entity, and "r" is the relationship between the two entities. Although these methods have achieved particular success, these models cannot express the complex relationships between entities [10] and cannot exploit the rich topological semantic information between entities [7], resulting in insufficient entity representation.

GNN-Based Representation. Benefiting from the rapid advancements in Graph Neural Networks (GNN) [17], especially in the representation of graph-structured data, GNN-based methods can utilize topological information to show promising performance in entity alignment. For example, GCN-Align [18], HMAN [19], RDGCN [20], and HGCN [21]. In contrast to models based on translation, GNN-based models contain the topological information of the entity and train the GNN to integrate the entity of each KG, and attributes are embedded in a low-dimensional vector space. This approach can aggregate neighborhood structure information to enrich entity vector representations.

However, most existing GNN-based entity alignment methods directly treat multi-relationship knowledge graphs as single relationships and rarely consider the relationship types between entities, resulting in insufficient and inaccurate entity representation. Although some methods consider complex multiple relationships between entities, they are mainly oriented to explicit relationships and ignore potential relationships between entities. Obviously, it is not enough to only consider explicit relationships. In real KGs, the relationships of each entity are intricate and can be connected to other entities through multiple relationships, even in the absence of direct relationships. For example, as shown in Fig. 1, entity "B" within KG1 establishes connections with various entities through a diverse array of relationships, thereby encapsulating a wealth of information within the entities' topological structure. Moreover, entity "B" maintains four distinct relationships with entity "F". However, the significance of these four relationships varies. Notably, KG1 does not exhibit a direct link between entity nodes "A" and "F". However, through the intermediary entity "B" potential associations can be inferred, further influencing the representation of entities.

To address the above issues, this paper proposes a graph convolutional neural network (MFLM-GCN) based on multi-relation fusion and latent-relation mining, which performs entity alignment by making full use of the topological semantic information of entities. The experimental results demonstrate that our proposed method can achieve superior entity node representation, leading to enhanced performance in the task of entity alignment. Specifically, we first connect two knowledge graphs through seed entities to promote information dissemination between graphs. Second, we exploit the local isomorphism between seed entities and graph random walks to capture meaningful latent relationships. Third, we propose an explicit and implicit multi-relation fusion method for the preliminary representation of entities, which can aggregate relationships in a local and global manner. Fourth, to further explore potential relationships between entities, we adopt a multi-head attention mechanism to generate multiple complete association graphs representing potential correlations between entities and build dense connection layers to obtain multi-branch representations of entities. Furthermore, to improve the aggregation performance of densely connected layers, we prune all correlation graphs with the previous graph random walk results before information aggregation. Finally, we use a linear layer to fuse entity features, obtain the final entity embedding, and complete entity alignment. The main contributions of this article include the following:

- We propose a novel multi-relation fusion and latent-relation mining graph convolutional network (MFLM-GCN) to optimize entity embeddings by mining and fusing topological information to improve the overall EA performance.
- We propose an efficient latent relationship mining strategy, using local isomorphism, graph neural walking, and multi-head attention to mine deep potential relationships between entities.
- We propose a local and global multi-relation fusion method that can efficiently fuse explicit and implicit relationships between entities.
- We evaluated the performance of MFLM-GCN. The results show that MFLM-GCN significantly outperforms seven baseline models in terms of Hit@1, Hit@10, and MRR indicators on the DBP15K and DWY100K datasets.

The remainder of this paper is structured as follows. Section 2 provides an overview of related work in the field of entity alignment. Section 3 illustrates the proposed neural network MFLM-GCN. The results of the experiments are showcased in Section 4, while Section 5 provides the concluding remarks of this paper.

2. Related work

2.1. Traditional methods based on translation

Traditional translation-based knowledge graph embedding methods [14,15,22,23] usually assume that each entity has sufficient training triples. This type of method maps entities and relations together into a low-dimensional vector space, while keeping the model parameters simple, showing excellent scalability and being able to effectively handle the embedding requirements of large-scale knowledge bases.

In knowledge graph representation, TransE [14] and its variant models [15,16] ascertain relationship and entity embeddings by considering relationships as equivalent to the vectorial translations from the head entity to the tail entity in vector space. For instance, TransH [15] represents relations as hyperplanes and conducts translation operations, assigning distinct hyperplanes and relation vector representations to each relation. Conversely, TransR [16] establishes embeddings in separate spaces for entities and relations, training embeddings through the projection of entities into corresponding relation spaces. This progressive augmentation of relational modeling intricacy significantly enhances knowledge graph representation, particularly concerning scenarios involving multi-mapping relations.

Furthermore, the earliest semantic matching model MTransE [22] employs TransE to embed entities and relations in different languages into distinct independent vector spaces and provides the transformation of each embedding vector to its cross-lingual counterpart in other spaces. In contrast, IPTransE [24] integrates multiple knowledge graphs into a unified semantic space. This integration is achieved by imposing constraints on the vector representations of alignment seeds, ensuring consistency across diverse knowledge graphs. SEA [23] enhances alignment by incorporating cycle consistency constraints into the loss function. Additionally, it leverages unlabeled data to obtain more accurate node vector representations. BootEA [25] places significant emphasis on addressing the challenge of having fewer pre-aligned entity pairs in the entity alignment process. It employs an iterative method to continuously select potential entity pairs for training. Nevertheless, the aforementioned methods disregard the topological structure information within the knowledge graph, consequently failing to acquire more enriched entity vector representations.

Beyond the structural information of the knowledge graph, the inclusion of semantic features like entities, relationships, and attributes also serves to elevate entity representation. Various methods [22] aim to enrich entity semantics by integrating a spectrum of diverse knowledge sources. JAPE [26] employs the TransE model for entity representation and employs Skip-gram [27] for acquiring attribute representations. Rooted in the premise that entities manifesting akin attributes tend to exhibit a heightened likelihood of equivalence, this approach extends attribute similarity to augment the semantic essence of entities. In addition, AttrE [28] introduces distinct attribute value representations to facilitate the acquisition of vectorized entity representations. Notably, it can autonomously establish alignments across datasets replete with diverse attribute values, obviating the need for preemptively aligned entity pairs. MultiKE maximizes the utilization of relational triples, attribute triples, and entity name information. It encodes these information sources separately, integrates them efficiently, and generates an entity representation that amalgamates multi-view information for the purpose of entity alignment.

2.2. Graph representation methods based on GCN

In terms of exploration in the field of topological structure, GNN [17] have garnered significant attention and research interest, achieving remarkable accomplishments in various domains such as node classification [29], graph classification [30], and link prediction [31]. Notably, Graph Convolutional Networks (GCN) [32] have harnessed node interconnectivity to capture comprehensive and localized graph-embedded information, thus elevating the caliber of node representations.

GNN-based models possess significant prowess in capturing intricate graph structures, rendering them well-suited for the task of aligning entities by representing knowledge graph nodes and relationships. For example, GCN-Align [18] pioneering the use of GNN for entity alignment, leverages a pre-aligned set of entities to capture varying degrees of entity similarity through neighbor aggregation across two GCN layers. MuGNN [33] focuses on learning alignment-oriented KG embeddings. It achieves this by leveraging multiple channels, allowing for the joint execution of knowledge inference and entity alignment. This approach aims to address the heterogeneity inherent in KG structures. AttrGNN [34] partitions a comprehensive KG into four subgraphs based on attribute values. It employs BERT [35] to derive the initial features for each attribute value. GMNN and NMN [36] approach the EA task as a graph-matching problem. In this framework, the alignment of entities is achieved by assessing the similarity of their respective subgraphs. AliNet [37] uses the gating mechanism to aggregate neighbors to expand the overlap of adjacent structures. It also proposes a relation loss to improve the representation of entities. However, these methods ignore the multiple relationships that often exist between two entities in the real world, resulting in insufficient and inaccurate entity representation.

The authentic graph data typically forms heterogeneous graphs [38], where node representation is shaped by neighboring nodes and their connecting relationships. To effectively capture the diverse semantic associations between nodes of distinct types and various kinds of edges, and to more precisely portray the complexities of real-world correlations, Relational Graph Convolutional Networks (RGCN) [39] improve GCN to address the impact of different edge relationships on nodes within the graph structure. This approach models [40,41] relational data, facilitating the transition from a homogeneous graph representation to a representation that embraces graph heterogeneity.

Inspired by RGCN, some entity alignment methods [20,21,42,43] fully consider the graph structure information and pay attention to the relationship information. For example, HGCN [21] considered the relation information in the entity alignment task and jointly learned entity and relation representations. Afterward, RDGCN [20] and DNCN [11] construct a dual relation graph for embedding learning, aiming to capture neighborhood relations effectively. ERGCN [42] models the connections between a pair of relations by constructing quadruples, facilitating the determination of their neighborhoods. RHGN [43] encompasses a specialized convolutional layer gated by relational attributes, engineered to discern and delineate the unique roles of both relations and entities within the knowledge graph. RREA utilizes relational reflection transformation to efficiently derive relation-specific embeddings for individual entities. RE-GCN [44] combines the original graph convolution with an innovative triplet graph convolution to obtain both relational and entity embeddings. In addition to evaluating neighboring nodes during the process of matching neighborhoods, RNM [45] exploits the constructive synergy between entity alignment and relation alignment in a semi-supervised manner. GEM-GCN [46] jointly optimizes node and edge embeddings for goal-driven goals. RAGA incorporates the self-attention mechanism to disseminate entity information to relations and subsequently aggregates relation information back to entities.

To enhance the outcomes of entity alignment, HMAN [19], EAMI [47], and BNGNN [48] seamlessly combine the topological structural information of a knowledge graph while also providing additional semantic and string information about relationships and attributes.

Compared with the above methods, we aggregate local and global neighbor relation information through multi-relational fusion. Multi-head attention is then employed to capture potential associations between entities and generate multiple fully-associated graphs. Second, a dense connection layer is introduced to enhance the information transfer process between sub-layers, preserving the topological information of the original graph. Our model improves entity representation by capturing multi-relationship information among entity nodes and integrating their latent features.

3. Proposed method

The model architecture of MFLM-GCN is shown in Fig. 2, which contains knowledge graph preprocessing, multi-relation fusion, latent-relation mining, and entity alignment. (1) Knowledge graph preprocessing: We construct the connection between the two knowledge graphs based on the seed entities and use the local isomorphism of the seed entities and the random walk of the graph to enhance the connection between entities. (2) Multi-relation fusion: We first aggregate each relationship in a local manner, and then aggregate all relations in a global manner to obtain an entity representation based on multiple relationships. (3) Latent-relation mining: We first use multi-head attention to generate multiple fully related graphs and use the graph random walk results to prune them. Then, we build a dense connection layer to capture the deep latent relationships of entities and obtain entity representations based on latent relationships. (4) Entity alignment: We use a linear fusion layer to fuse multiple representations of entities to obtain the final embedding of entities and achieve entity alignment.

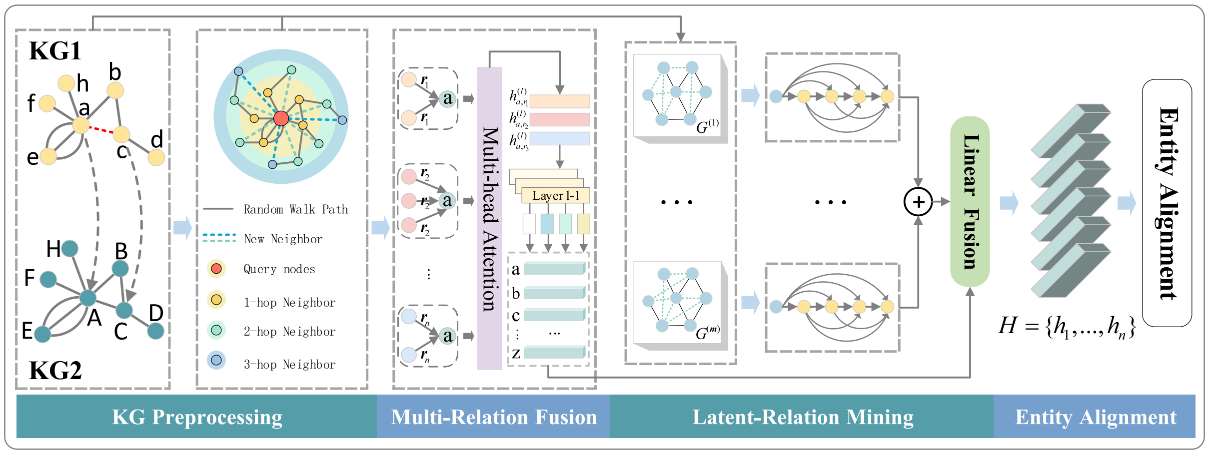


Fig. 2. The framework diagram of the entity alignment method MFLM-GCN proposed in this paper mainly includes four parts: knowledge graph preprocessing, multi-relation fusion, latent-relation mining, and entity alignment.

3.1. Problem definition

Entity alignment involves identifying corresponding entities in two distinct knowledge graphs. A knowledge graph is represented as $G = (E, R, T)$, where E is the set of entities, R is the set of relations, and T is the set of triplets. In this paper, we denote the source knowledge graph as $G_1 = (E_1, R_1, T_1)$, the target knowledge graph as $G_2 = (E_2, R_2, T_2)$, and the seed-aligned entity set $S = \{(u, v)|u \in E_1, u \in E_2, u \leftrightarrow v\}$, denoting that the entity u in the knowledge graph G_1 is the same entity as v in the knowledge graph G_2, where the $\leftrightarrow$ representations equal. Our entity alignment model acquires alignment information by matching entities with seeds, aiming to identify additional entities in the target knowledge graph that align with entities in the source knowledge graph.

3.2. Knowledge graph pre-processing

3.2.1. Graph fusion and isomorphism

Aligned entities refer to the same thing in real life, and their vector representations should be very similar. We can connect the two knowledge graphs through the aligned seed entities given in the dataset, which can enhance the vector representation of the aligned entities and enable the model to learn richer information, thereby promoting the alignment of other entities. Therefore, to promote information sharing and fusion between the two knowledge graphs, we define an align tag to represent the alignment relationship and merge it into the triplet T. The formula is as follows:

$$T \leftarrow T \cup \{(u, align, v)|(u, v) \in S, align \in R\} \tag{1}$$

In addition, there are often differences in the local structure between seed entity pairs in different knowledge graphs. This local topological difference affects the consistent representation between pairs of entities when the model encodes and represents nodes. Therefore, we exploit existing information to improve the isomorphism of two different knowledge graphs.

As shown in Fig. 2, node A in KG_2 and node a in KG_1 are aligned entities, and similarly, node C and node c are aligned entities. In KG_2, there is a relational connection between node A and node C. Through relational inference, it can be known that node a and node c in KG_1 also have the same relationship, and a new edge is constructed between entities with such a relational inference mapping. To increase the isomorphism of the graph, the formula is as follows:

$$\begin{aligned} &(A, R, C) \rightarrow (a, R, c) | A \leftrightarrow a, C \leftrightarrow c \\ &(A, R, C) \in G_1; (a, R, c) \in G_2 \end{aligned} \tag{2}$$

The reconstructed graph contains a large number of topological features. Based on the process of graph fusion and isomorphism, we get the original graph $G^{(o)}$ and its corresponding adjacency matrix $A^{(o)}$.

3.2.2. Graph random walk

In the knowledge graphs, the relationship distribution of entities conforms to the long-tail effect. That is, a small number of entities have many relationships, while most entities have only a small number of connections. Therefore, the structural features of most entities suffer from sparsity problems. In order to mine richer structural features in the graph to alleviate the feature sparseness problem of long-tail entities, we introduced the graph random walk method. Graph random walk simulates random walk paths between nodes, obtains the accessibility and path features between node pairs, and thus mines potential relationship entities in the graph. Then, by aggregating these potential relationship entities, the feature sparseness problem of long-tail entities can be effectively alleviated.

Furthermore, for a comprehensive exploration of latent relations between entities, the dense connection layer is tasked with processing the complete associative graph. When the number of nodes of the entity becomes too large, the attention score matrix will reach unmanageable dimensions, seriously affecting the training effect of the model. The connectivity between entities can be enriched through a graph random walk. The adjacency matrix that introduces new neighbors can be used as the input of the dense connection layer, replacing the full-associative graph. This substitution not only significantly reduces resource requirements but also enhances overall performance.

Specifically, the graph random walk computes scores between the query node and other nodes in the graph using an affinity scoring algorithm (e.g., RWR) tailored for entity nodes. Subsequently, it organizes the other nodes in descending order based on these scores, forming a sequence. Initially, 1-hop neighbors are extracted, and then a normalized summation is performed, selecting nodes in descending order until reaching a cumulative score of λ $(0 \le \lambda \le 1)$. The update process of neighbor nodes is shown in Fig. 3. This process yields a new adjacency matrix following graph random walk processing, as depicted by the following formula:

$$N_{e_i}^{GRW} = N_{(e_i,\text{1-hop})} \cup \{e_j \mid e_j \in \text{GRW}(e_i, f),\ e_j \notin N_{(e_i,\text{1-hop})}\} \tag{3}$$

where $N_{e_i}^{GRW}$ represents the entity neighbor set updated by random walk, $N_{(e_i,\text{1-hop})}$ represents the first-order neighbors of the entity, e_j represents other entities selected by random walks that are not in the first-order neighbors of the entity, and $\cup$ represents the union operation of the set.

3.3. Multi-relation fusion

3.3.1. Relationship coefficient initialization

Entity pairs establish connections through relationships, with information transmission occurring along edges constructed by these

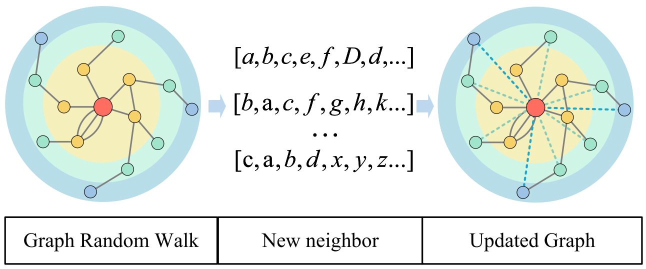


Fig. 3. Nodes exhibiting robust correlations with the target node are identified. This process results in the creation of new edge connections, leading to the formation of an updated graph.

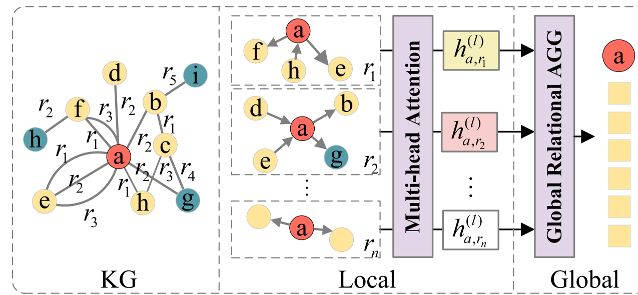


Fig. 4. Multi-relationship aggregation entity information process. First, aggregate entity information from each local relationship, then aggregate entity information from all global relationships.

relationships. Examining the graph structure based on relationship information, we compute a relationship coefficient for each relationship on the entity node. Subsequently, we assign a weight to each edge based on the relationship coefficient, following the formula below:

$$a_{i,r} = \frac{1}{C_r} \cdot \frac{AC_r}{\sum_{k \in R} AC_k} \tag{4}$$

$$a_{ij} = \sum_{r \in R_{ij}} a_{i,r} \tag{5}$$

where $a_{i,r}$ and C_r represent the relationship coefficient and the quantity of relationship r in the entity node e_i, respectively. and AC_r enotes the total count of relationship r in the entire knowledge graph. a_{ij} represents the weight between the entity nodes, obtained by summing the relationship coefficients of all edges between entity nodes e_i and e_j, R_{ij} are all relationship types between entities e_i and e_j.

3.3.2. Multiple relationship aggregation

Given the complexity and diversity of knowledge graphs, each entity node usually has multiple relationships with other entities. However, traditional GCN shows certain limitations in handling multiple relationships. These methods mainly infer the characteristics of the target node by aggregating limited neighborhood information through each relationship. This means that these methods can effectively propagate entity alignment information between graphs only when the pre-aligned relationships between two knowledge graphs are rich. Nevertheless, seed entity pairs between knowledge graphs are usually scarce. In this case, the ability of this per-relation-based neighborhood aggregation method to effectively propagate entity alignment information will be weakened.

In order to enhance the information aggregation ability of multi-relationship graph neural networks, we adopts a novel strategy — aggregating entity information based on local and global relationships. As shown in Fig. 4, the core of this strategy is to comprehensively consider the local and global impacts of entity relationships. On the one hand, it focuses on the relationship features of entities within local neighborhoods to capture their direct association information; On the other hand, it introduces a global relationship perspective to identify broader and more complex potential relationships between entities throughout the entire graph structure. By combining the local and global approaches, we can generate entity representations with global relationship awareness, thereby obtaining richer and more complete multi-relation information and effectively improving the performance of entity alignment.

In the process of local relationship aggregation, we expect to obtain more detailed relationship information for each entity node. Therefore, at each layer, for each relationship r, we aggregate the information of all related entities connected by the entity nodes under the relationship. At the same time, in order to further capture the complexity of local relations, we introduce a multi-head attention mechanism to more comprehensively extract local relation features from multiple dimensions:

$$h^{(l)}_{e_i,r,K} = \sum_{e_j \in N_r(e_i)} (a_{i,r} \cdot W^{(l-1)}_{r,K} \cdot h^{(l-1)}_{e_j,r}) \tag{6}$$

$$h^{(l)}_{e_i,r} = AGG^{(l)}_{local,r}(Concat(h^{(l)}_{e_i,r,1}, h^{(l)}_{e_i,r,2} \dots h^{(l)}_{e_i,r,K}) \tag{7}$$

where $N_r(e_i)$ represents the set of neighbor entity nodes connected to entity node e_i through relation r, K represents the number of heads of multi-head attention, $W_r^{(l-1)}$ represents the weight matrix of the K-th head under relation r and $Concat$ represents the concatenation of the outputs of multiple heads along the feature dimension.

Then, during the global relation aggregation process, we share the same weight matrix with the dense connection layer and the global relation aggregation. It is hoped that the entity node representation of layer l can learn the relationship information of the previous layer and simultaneously integrate all the relationship information of layer l. We define the global relationship aggregation of l-level entity nodes as:

$$h^{(l)}_{e_i} = h^{(l-1)}_{e_i} \cdot AGG^{(l)}_{global,r}(\sum_{r \in R} W^{(l)}_r \cdot h^{(l)}_{e_i,r}) \tag{8}$$

where $h^{(l)}_{e_i}$ represents the hidden vector of the entity node in the l layer, and $h^{(l)}_{e_i,r}$ is the local relation vector representation in the l layer under the relation r. We propose a global relational aggregator $AGG^{(l)}_{global,r}$, which learns the local relational information of layer l through the relational weight matrix $W_r^{(l)}$ and fuses the hidden entity node vectors of layer $l-1$ through splicing aggregation operators, e.g., concatenation, or Multi-layer Perceptron (MLP).

3.4. Latent-relation mining

3.4.1. Fully associated graph generation

The knowledge graph contains a large amount of topological structure information. Entity pairs with edge connections will have a direct impact on each other. Nonetheless, a potential influence may arise in scenarios where direct edge connections between entity nodes are absent. Existing methods may struggle to capture this latent relationship between entities. Therefore, we employ multi-head attention on the original pre-aligned knowledge graph, generating multiple attention adjacency matrices and obtaining a variety of fully associated pre-aligned knowledge graphs with different weights, thereby capturing potential relationships between all entity pairs. Furthermore, the scaled dot product attention mechanism is known for its efficiency in terms of speed and space utilization. Consequently, we employ this mechanism in our computations, and its formula is expressed as follows:

$$A^{(z)} = softmax(\frac{QW^Q_z \times (KW^K_z)^T}{\sqrt{d_z}})A^{(0)} \tag{9}$$

where z denotes the index of the multi-head attention and the fully associated graph, $z \in \{1, \dots, m\}$. $A^{(z)}$ stands for the adjacency weight matrix of the zth fully associated graph, and $A^{(0)}$ represents the initial adjacency weight matrix composed of a_{ij}. $Q \in R^{z\times d}$ and $K \in R^{z\times d}$ refer to the feature vectors of the entity nodes in the $l-1$-th layer. $W_z^Q \in R^{d\times d}$ and $W_z^K \in R^{d\times d}$ act as the linear transition matrices of Q and K, respectively. d_z is the dimension of the feature output.

$A^{(m)}$ obtained through multi-head attention results in a fully associated edge-weighted pre-aligned entity graph. The graph convolution model enables the capture of interactions between arbitrary nodes, facilitating the extraction of latent relationships between nodes. Multiple attention is given to obtaining multiple fully associated weighted graphs to capture different latent information with separate attention. Furthermore, this design tackles the constraints present in the original graph structure, offering potentially pertinent information between constituent utterances for subsequent stages of information propagation.

3.4.2. Information propagation

We obtain multiple fully associated graphs with edge weighting in the multi-head attention layer. However, directly propagating information on fully associated graphs has problems such as high computational complexity and excessive storage space. Therefore, to promote deeper information propagation while reducing computational complexity, we introduce dense connections and data augmentation mechanisms in information propagation, the detailed structure shown in Fig. 5.

First, we use a multi-head attention mechanism to capture the latent relationships between entities from multiple perspectives and generate multiple fully connected graphs. Then, we generate candidate nodes through the random walk step of the Formula. (3), and integrate the candidate nodes with the original neighbors to form a new set of neighbors. Finally, we prune the fully connected adjacency matrix to only retain the connection relationship between the entity and the first-order neighbors and the filtered candidate nodes, thereby reducing the computational complexity. For each attention head, we perform pruning to generate multiple enhanced graphs. We use the enhanced graph as the input of the densely connected layer, and promote cross-layer propagation of information through deep connections, thereby retaining more useful prior information and enhancing the expressiveness of the model.

The dense connections strategy establishes direct connections between the hidden vectors of the current layer and all previous layers when representing the entity nodes of the l layer. Compared with traditional GCN, in the case of dense connections, the node of the lth layer not only receives input from the $(l-1)^{th}$ layer but also integrates information from all the previous layers. Mathematically, the node feature representation generated by layer l in the attention head zth is defined as:

$$X_{e_i}^{(z,l)} = [h_{e_i}^{(0)}; h_{e_i}^{(z,1)}; \dots; h_{e_i}^{(z,l-1)}] \tag{10}$$

where $h_{e_i}^{(z,l-1)}$ represent the vertex features generated by the entity node e_i in the zth attention head through the $(l-1)^{th}$ sublayer, which obtained by the fully associated graph. $X_{e_i}^{(z,l)}$ denotes the connection between the initial entity feature vector under the dense sublayer l of the zth attention head and the hidden feature vectors $h_{e_i}^{(0)}$ of all previous dense sublayers.

The hidden node vector calculation of each sub-layer is related to the number z of fully connected edge weighted entity pre-aligned graphs $G^{(z)}$ generated by multi-head attention, where the zth fully associated graph $G^{(z)}$ is in layer l. The calculation formula in the sublayer is as follows:

$$h_{e_i}^{(z,l)} = AGG(\sum_{j=1}^{n} A_{ij}^{z} W^{(z,l)} X_{e_j}^{(z,l-1)} + b^{(z,l)}) \tag{11}$$

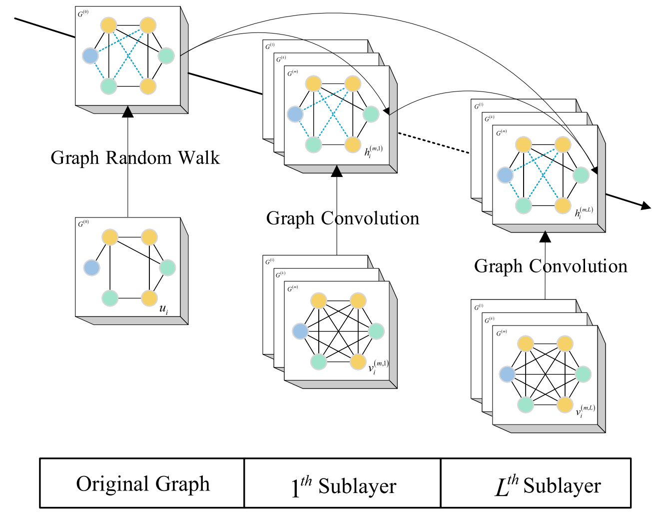


Fig. 5. Taking the pruned fully associated graph as input, the information transfer mechanism between each sub-layer of the dense connection layer is shown from left to right, as well as the aggregation process of graph convolution from bottom to top.

where A_{ij}^{z} represent the adjacency matrix, which correspond to the first fully connected edge weighted entity pre-alignment graph $G^{(z)}$, $X_{e_j}^{(z,l-1)}$ denotes the feature vector representation of the entity node e_j of the zth attention head under the dense sublayer $l-1$, and $b^{(z,l)}$ represents the offset value, $W^{(z,l)}$ denotes the corresponding weight matrix, and AGG stands for an aggregation function in the given context.

In the convolution process of traditional GCN, the hidden vector dimension is greater than or equal to the input vector dimension. Differing from this, the dense connection strategy dictates that the hidden vector of each sublayer is defined as $d_{hidden} = \frac{d}{L}$, where d is the dimension of the input vector, and L is the number of layers in the dense connection layer. It is evident that as the number of layers L increases, d_{hidden} diminishes, thereby enhancing parameter efficiency. Ultimately, the results of each sublayer are concatenated to form the final output, with the dimension of the output vector matching that of the input vector. The dense connection layer combines shallow and deep features through cross layer feature concatenation, while reducing the hidden dimension of each layer. This design not only achieves effective fusion of multi-level features while maintaining controllable total parameter quantity, but also enhances information flow and avoids performance degradation of deep networks caused by information dilution.

3.4.3. Linear fusion

Through the dense connection layer, we can get the hidden vectors $h_{e_i}^{(z,L)}$ of z entity nodes learned by multi-head attention. In the multi-relational fusion layer, we get the final relationship-aware vector $h_{e_i}^{(L)}$ of entity nodes. The result is shown in the formula:

$$h_{e_i}^{(L)} = h_{e_i}^{(L-1)}.AGG_{global,r}^{(l)} \tag{12}$$

$$h_{e_i}^{(z,L)} = [h_{e_i}^{(0)}; h_{e_i}^{(z,1)}; \dots; h_{e_i}^{(z,L-1)}] \tag{13}$$

We design a linear fusion layer to fuse the output of the dense connection layer and the multi-relational fusion layer to obtain the final representation of entity nodes. The linear fusion formula is as follows:

$$h_{e_i} = \beta(W_{line} h_{e_i}^{(z,l)} + b_{line}) + (1-\beta) h_{e_i}^{(L)} \tag{14}$$

where b_{line} is the bias item of the linear fusion formula, and W_{line} is the linear weight matrix. β is a hyperparameter used to balance the results.

Table 1
Analysis of the DBP15K dataset.

Datasets		DBP15K				
		Entities	Relations	Attr′	Rel Tri′	Attr Tri′
JA-EN	Japanese	65 744	1299	5882	164 373	354 619
	English	95 680	2096	6066	233 319	497 230
FR-EN	French	66 858	1379	4547	192 191	528 665
	English	105 889	2209	6422	278 590	576 543
ZH-EN	Chinese	66 469	2830	8113	153 929	379 684
	English	98 125	2317	7173	237 674	567 755

′ Abbreviated form of Attributes, Relation and Triples.

Table 2
Analysis of the DWY100K dataset.

Datasets		DWY100K				
		Entities	Relations	Attr′	Rel Tri′	Attr Tri′
DBP-YG	DBpedia	100 000	302	334	428 952	451 646
	YAGO3	100 000	31	23	502 563	118 376
DBP-WD	DBpedia	100 000	330	351	463 294	381 166
	Wikidata	100 000	220	729	448 774	789 815

′ Abbreviated form of Attributes, Relation and Triples.

3.5. Entity alignment

Entity alignment aims at the task of identifying equivalent entities in two distinct knowledge graphs and establishing connections between them for knowledge fusion. Following the acquisition of vector representations for all entity nodes via MFLM-GCN, we assess the similarity between entities by examining the distances between their respective vectors in a graph. Entities with smaller vector distances are more likely to be aligned. We arrange each entity node in the target graph based on vector distances and designate the one with the smallest distance as the candidate for alignment. The distance formula between two entities is specifically expressed as:

$$d(e_i, e_j) = \|h_{e_i} - h_{e_j}\|_{L_1} \tag{15}$$

where e_i is from G_1 and e_j is from G_2.

During the model training process, the entity vectors obtained through model learning are utilized to compute the loss value. The objective is to optimize model training by minimizing the margin-based ranking loss function with a margin of τ_{loss}, aiming to achieve the optimal performance in entity alignment. The formula for the loss function is:

$$\tau_{loss} = \sum_{(e_i,e_j)\in S} \sum_{(e_i{}',e_j{}')\in S'} [d(e_i, e_j) + \gamma - d(e_i{}', e_j{}')]_+ \tag{16}$$

where $(e_i, e_j) \in S$ denotes the entity alignment seed. During the model learning process, we aim for the distance between aligned entity nodes to be minimized, while the distance between the same entities should be maximized. To achieve this, we construct a batch of negative sample sets $S' = \{(e_i', e_j), (e_i, e_j')|e_i, e_i' \in G_1, e_j, e_j' \in G_2\}$ to be used in training the model. The notation $[\cdot]_+$ represents $max\{0, \cdot\}$, and γ denotes the margin hyper-parameter.

3.6. Complexity analysis

In order to comprehensively evaluate the performance of the model, we conducted a time complexity analysis on the four key components of MFLM-GCN, namely, Graph Fusion and Isomorphism, Graph Random Walk, Multi-Relation Fusion, and Latent-Relation Mining. First, in the Graph Fusion and Isomorphism part, its complexity mainly comes from the traversal of seed pairs and all their relations, so the complexity is $\mathcal{O}(|E| \times |R|)$. Second, in the Graph Random Walk part, each entity performs a random walk, and each step needs to traverse all neighbor nodes. Therefore, the complexity of this part is $\mathcal{O}(|E| \times |R| \times |E|)$. Next, in the Multi-Relation Fusion part, its complexity mainly depends on local relationship aggregation and global relationship aggregation. Specifically, the complexity of local relation aggregation is $\mathcal{O}(|E|\times|R|)$, while the complexity of global relation aggregation is also $\mathcal{O}(|E|\times|R|)$. Therefore, the total complexity of this part is $\mathcal{O}(|E|\times|R|)+\mathcal{O}(|E|\times|R|) = \mathcal{O}(|E| \times |R|)$. Finally, in the Latent-Relation Mining part, its complexity mainly depends on the process of Fully Associated Graph Generation, which has a complexity of $\mathcal{O}(|E| \times |E|)$. In summary, the total time complexity of MFLM-GCN is $\mathcal{O}(|E| \times |E| \times |R| + |E| \times |R|)$.

4. Experiment

4.1. Datasets and baselines

Our experiments are conducted on two real-world multilingual data sets. Tables 1 and 2 provide statistical summaries for each dataset. The quantitative information includes statistics on entities, relations, attributions, relation triples, and attribute triples.

DBP15K is a cross-lingual entity alignment dataset with three KG pairs: ZH-EN, JA-EN, and FR-EN. In [26], the authors extracted 15,000 links from DBpedia, annotating them as synonymous or non-synonymous relations.

DWY100K [25] is a cross-KG alignment dataset comprising three networks: Wiki, YAGO, and DBP15K. It offers three sets of alignment tasks with varying difficulty levels. Each task maps entities in one network to entities in another network.

We conduct a comparative analysis of MFLM-GCN against seven representative baseline models. This includes three translation-based models and five GNN-based models, respectively.

- TransE [14] projects both entities and relationships into a low-dimensional vector space.
- MTransE [22] maps entities from each language into distinct low-dimensional vector spaces and computes transformations that map them into the spaces of other languages.
- IPTransE [24] integrates relational attribute information by embedding relations in entity representation.
- GCN-Align [18] conducts graph convolution operations across different knowledge graphs to capture relationships and semantic information between entities.
- MuGNN [33] treats multiple knowledge graphs as a collection of graphs, utilizing multi-graph joint training and multi-layer graph convolution operations.

Table 3
Our proposed model is evaluated by comparing its performance with 8 baselines and LSM-GCN on DBP15K. We evaluate performance using Hit@k and MRR, which represent the top k accuracy scores and average ranking, respectively. Those that perform best in vertical contrast are highlighted in bold.

	$DBP15K_{JA-EN}$			$DBP15K_{FR-EN}$			$DBP15K_{ZH-EN}$		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
TransE	20.13	46.92	0.31	21.36	48.01	0.33	18.97	44.68	0.29
MTransE	25.01	57.24	0.36	24.79	57.74	0.36	20.92	51.23	0.31
IPTransE	36.70	69.30	0.47	33.30	68.50	0.45	40.60	73.50	0.51
GCN-Align	41.54	74.64	0.55	40.43	76.67	0.54	40.93	72.88	0.54
MuGNN	50.10	85.70	0.62	49.50	87.00	0.62	49.40	84.40	0.61
AliNet	54.90	83.10	0.64	55.20	85.20	0.65	53.90	82.60	0.62
HMAN	55.76	86.10	0.67	55.03	87.61	0.66	56.14	85.73	0.67
GALA	56.83	81.78	0.65	58.09	84.06	0.66	56.33	81.11	0.65
LSM-GCN	57.18	88.39	0.69	59.83	89.98	0.71	57.21	87.21	0.68
MFLM-GCN	**63.97**	**90.10**	**0.73**	**68.13**	**91.89**	**0.77**	**63.17**	**88.29**	**0.72**

- AliNet [37] employs adaptive feature exchange and multi-layer attention mechanisms.
- HMAN [19] integrates both global and local information, incorporating attention mechanisms of different scales.
- GALA [7] aligns the entities by forcing their global features to match with each other and progressively updating the entity embeddings by aggregating local information from the other network.

Our experiments are conducted on NVIDIA A800 80 GB. We utilized the BERT model to obtain semantic vectors for entities, each with a dimension of 256. These vectors were then concatenated to form entity node feature vectors with a dimension of 512. In our architecture, we set the multi-head attention count for Multi-Relation Fusion to 2 and for Latent-Relation Mining to 4. The number of sub-layers in the dense connection layer was also configured to be 4. For each positive sample, we generated 20 negative samples ζ. During model training, the learning rate was set to 0.003. We used 30% of the dataset as the training set, and the model was trained for 300 epochs. Additionally, the model was saved every 50 epochs.

4.2. Evaluation metrics

Consistent with prior studies, we adopt Hit@k and MRR as the evaluation metrics for our model. A detailed description of each metric is provided below.

Hit@k: Entities are sorted based on the distances between their codes, and the proportion of correctly aligned entities within the top-k positions is calculated. We have chosen commonly used values of 1 and 10 for k. Hit@1 evaluates the model's precision in representing triples exactly and accurately. Hit@10 provides a more lenient assessment, considering correctly aligned entities within the top 10 positions as valid.

MRR: The Mean Reciprocal Rank (MRR) serves as a comprehensive metric for evaluating search algorithms globally. It involves the summation of the reciprocal values of correctly aligned entity rankings, followed by averaging. Its calculation formula is as follows:

$$MRR = \frac{1}{N} \sum_{e_i \in S} \frac{1}{rank_{e_i}} \tag{17}$$

where $rank_{e_i}$ denotes the position of the correctly aligned entity upon sorting based on the distance of entity codes, and N represents the total number of entities in the test set S.

4.3. Qualitative comparison

Tables 3 and 4 present the descriptive statistics, including means and standard deviations, pertaining to baseline models' performance evaluation on both the DBP15K and DWY100K datasets. An observation discernible from these tables is the compelling performance of the MFLM-GCN model across various language pairs, surpassing other comparative models in the evaluations.

Table 4
Performance of MFLM-GCN in Handling Large-Scale and Diverse Knowledge Graphs such as DWY100K also made significant progress in model design and algorithm optimization.

	DBP-YG			DBP-WD		
	Hit@1	Hit@10	MRR	Hit@1	Hit@10	MRR
IPTransE	29.73	55.81	0.37	34.92	63.86	0.45
GCNAlign	59.72	83.85	0.68	50.64	77.28	0.60
MuGNN	73.91	93.76	0.81	60.41	89.48	0.70
AliNet	78.65	94.32	0.84	69.09	**90.89**	0.77
MFLM-GCN	**80.80**	**95.73**	**0.86**	**69.11**	87.59	**0.77**

(1) Analysis of DBP15K indicates that the performance of translation-based models generally falls short compared to GNN-based models. This inferior performance is attributed to the challenges faced by translation-based methods in adequately representing relations, particularly in complex adjacency and ring structures. This observation aligns with the consensus in most related methodologies. In comparison to conventional GNN-based models like GCN-Align, which uses a standard GCN for entity embedding, MFLM-GCN demonstrates a significant 22.3% improvement in Hit@1 on the JA-EN dataset. This highlights the efficacy of MFLM-GCN's entity convolution function in revealing latent association information, thereby improving entity alignment. Furthermore, MFLM-GCN's notable performance gains over LSM-GCN emphasize the effectiveness of its novel graph random walk and multi-relation fusion modules in entity alignment tasks.

(2) Due to the fully connected graph module, LSM-GCN faces a quadratic increase in memory usage as the knowledge graph expands, hindering its efficiency in processing large-scale graphs. In contrast, MFLM-GCN, enhanced by its graph random walk module, effectively overcomes this limitation and demonstrates strong performance on large-scale knowledge graphs. According to data from DWY100K, translation-based models like IPTransE exhibit a marked performance drop on large-scale knowledge graphs. In contrast, GNN-based models show more pronounced performance improvements in such environments, as seen in comparisons with DBP15K. This improvement can be attributed to the richer topological structure and relationship information in large-scale knowledge graphs, which GNN-based models effectively leverage. Notably, MFLM-GCN also outperforms other models in most metrics. On the DBP-YG dataset, MFLM-GCN achieves Hit@1 and Hit@10 scores of 80.80 and 95.73, respectively, and an MRR index of 0.86. This underscores the superiority of our method in processing large-scale datasets.

These findings collectively affirm the efficacy of MFLM-GCN in addressing entity alignment challenges and the importance of its innovative modules.

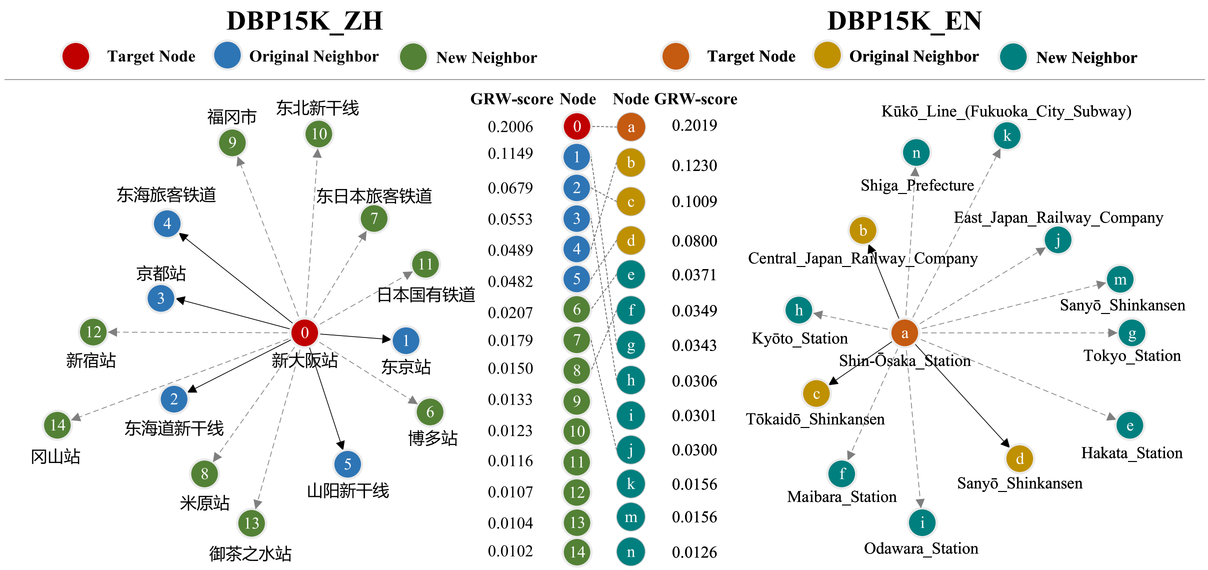


Fig. 6. Illustrative instances of the utility of graph random walks are demonstrated on the DBP15K_ZH and DBP15K_EN datasets. Specifically, target nodes exhibit connectivity to neighboring nodes with elevated GRW scores through stochastic traversal within the graph. This traversal encompasses both the exploration of original neighbor nodes and the discovery of novel neighbor nodes. The connections to original neighbor nodes are denoted by solid lines, while those to new neighbor nodes are indicated by dashed lines, delineating the distinct relationships established during the random walk process.

Table 5
The performance evaluation of various convolutional methods on the DBP15K dataset.MFLM-MR represents MFLM-GCN that only uses the MR module.

Datasets	JA-EN		FR-EN		ZH-EN	
Method	Hit@1	MRR	Hit@1	MRR	Hit@1	MRR
GCN	41.54	0.55	40.43	0.54	40.93	0.54
GAT	48.02	0.60	50.49	0.63	46.44	0.58
R-GCN	49.98	0.62	52.17	0.65	49.41	0.61
LSM-GCN	57.18	0.69	59.83	0.71	57.21	0.68
MFLM-MR	**60.40**	**0.70**	**64.52**	**0.91**	**58.10**	**0.69**

4.4. Ablation studies

To demonstrate the ability of MFLM-GCN to exploit relationships, the effectiveness of each design in each method module, and the spatiotemporal optimization brought by graph random walks, we constructed three ablation experiments on DBP15K.

The Ability to Exploit Relationships. To assess the efficacy in leveraging relations among various convolutional models, we substitute MFLM-MR with re-tuned variants of GNN, namely, GCN [32], GAT [49], and R-GCN [39], all configured with identical parameters. The results presented in Table 5 reveal that MFLM-MR outperforms GCN and GAT by 18.86% and 12.38%, respectively. This discrepancy arises from the fact that both GCN and GAT neglect relationships. While R-GCN encounters challenges in fully harnessing relationship information, our proposed method excels by achieving a notable improvement of 10.42% over R-GCN. Moreover, the superior performance of R-GCN compared to GCN and GAT underscores the crucial role of relationships in entity representation. In comparison to our previously proposed LSM-GCN, MFLM-GCN exhibits further enhancements.

The Effectiveness of Each Module. MFLM-GCN incorporates four key components to capture diverse aspects of information present in KGs: (1) *Graph Random Walk* (GRW), (2) *Multi-Relational Aggregation* (MR), (3) *Multi-Head Fully Connected Graph* (MH), (4) *Dense Connected Layer* (DC). We perform an ablation study on each component of MFLM-GCN, evaluating their individual influences on performance. The corresponding results are presented in Table 6.

- The 1st and 2nd rows show the influences of multi-relational aggregation (MR) on EA performance. Compared with only using GCN for entity alignment, MR can improve the Hit@1 by 18.86%. This improvement stems from the importance of relational information in the knowledge graph.
- The 1st and 3rd rows show the influences of multi-head attention (MH) and densely connected layers (DC) on capturing latent relational information. We can see that using MH and DC increases the accuracy by 15.64% compared to GCN alone. This indicates that the fully connected graph generated by MH convolved by DC can improve performance, capture potential relationship information between entities and the deeper information contained in all convolutional network hidden layers.
- The 3rd and 4th rows show the influences of MR on MFLM-GCN, which brings 6.66% accuracy improvement. This enhancement underscores the effectiveness of MR in fully leveraging relational dynamics, thereby enabling a more robust learning of entity representations.
- The 4th and 5th rows show the influences of employing graph random walk (GRW). Compared with no GRW, the accuracy is improved by 0.13%, which shows that the GRW module improves the alignment performance to a certain extent. The GRW is more about reducing the memory pressure caused by the fully connected graph.

The Spatiotemporal Optimization Brought by Graph Random Walk. The goal of the graph random walk is to randomly sample adjacent nodes and selectively identify neighbors exhibiting high structural correlation, thereby enhancing the neighbor set. To explore the effectiveness of our model in relational representation, as shown in Fig. 6, we took two nodes from the $DBP15K_{ZH-EN}$ graph as examples and visualized the experimental results of GRW. The analysis reveals two phenomena: neighbors that align with nodes in ZH were sampled for nodes in EN, and newly expanded neighbors in the EN and ZH graphs have aligning nodes. We can know that GRW effectively identifies relevant neighbors, alleviating the sparsity of the knowledge graph structure.

Moreover, the results of the graph random walk, combined with a fully connected graph, significantly reduce resource requirements and improve performance while thoroughly exploring potential relationships between entities. As shown in Table 7, on the $DBP15K_{ZH-EN}$

Table 6
Ablation experimental performance of MFLM-GCN on the DBP15K dataset with different components.

Components				JA-EN		FR-EN		ZH-EN	
GRW	MR	MH	DC	Hit@1	MRR	Hit@1	MRR	Hit@1	MRR
–	–	–	–	41.54	0.55	40.43	0.54	40.93	0.54
–	✓	–	–	60.40	0.70	64.52	0.91	58.10	0.69
–	–	✓	✓	57.18	0.69	59.83	0.71	57.21	0.68
–	✓	✓	✓	63.84	0.73	67.21	0.76	62.17	0.71
✓	✓	✓	✓	63.97	0.73	68.13	0.77	63.17	0.72

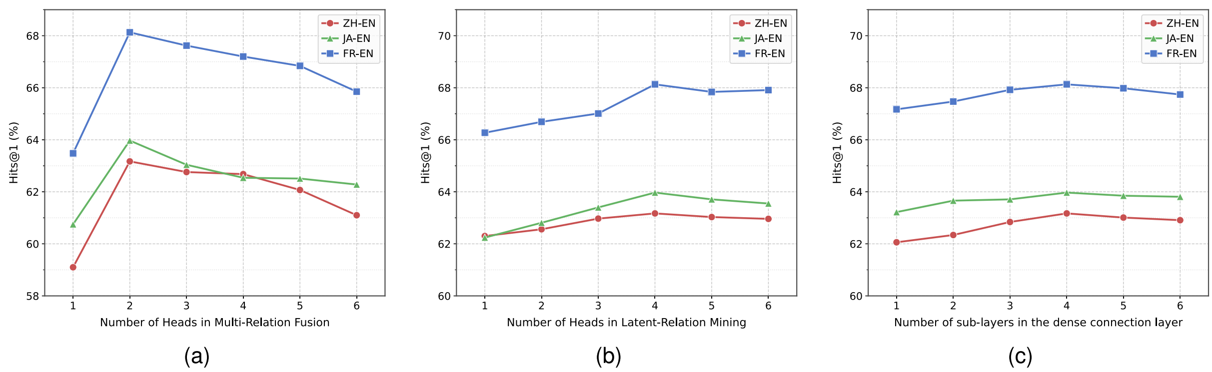


Fig. 7. The impact of different numbers of multi-relation fusion attention heads (a), latent-relation mining attention heads (b), and sub-layers of densely connected layers on MFLM-GCN.

Table 7
Performance of graph random walk.

Method	Times (s)	Space (M)
MFLM	5.761	9707
MFLM(w/o grw)	10.354	32 361

dataset, MFLM saved nearly 0.8 × execution time and over 2.4 × memory space after using GRW, making it feasible for entity alignment tasks in larger-scale knowledge graphs.

The above experiments demonstrate that using a graph random walk to sample neighbors with high structural correlation can better obtain entity representations and reduce memory resource requirements.

4.5. Parameter experiment

In this section, we conduct parameter experiments on the number of attention heads in multi-relation fusion, the number of attention heads in latent relation mining, and the number of sub-layers in the densely connected layers to analyze their effects on the model.

The number of heads in multi-relational fusion. In our investigation, we systematically examine diverse configurations for the number of attention heads in the context of multi-relation fusion, elucidating its influence on the depth of potential relationships explored during the entity alignment process. Illustrated in Fig. 7(a), we varied the number of attention heads from 1 to 6 across the three datasets in DBP15K, aiming to discern its impact on the model's capacity to discern varied semantic relationships between entities. Our scrutiny of the alignment outcomes reveals that the optimal configuration, in terms of model performance, is observed when employing 2 attention heads.

The number of heads in latent-relation mining. We experimented with the number of multi-head attention heads in latent-relation mining on three datasets of DBP15K, ranging from 1 to 6. The experimental results in Fig. 7(b) show that as the number of multi-head attention heads increases, the model performance shows a trend of first increasing and then decreasing. Specifically, when the number of heads is 4, the model performance reaches the best state, which indicates that the model can most effectively capture the diversity of latent relationships at this time. However, when the number of heads exceeds 4, the model performance begins to decline slightly. The reason for this phenomenon is that when the number of heads is small, the model may not be able to fully capture the complexity of the latent relationship; when the number of heads is too large, the model may introduce redundant information, resulting in performance degradation.

The number of sub-layers in densely connected layers. Similarly, we experimented with the number of sub-layers of densely connected layers in the three language directions of the DBP15K dataset, ranging from 1 to 6. From the data in Fig. 7(c), it can be seen that the model performance first increases and then decreases with the increase in the number of sub-layers, and the best state appears when the number of sub-layers is 4. However, it is worth noting that even under different parameter values, the model effect does not change much, and the model performance always remains at a high level. This phenomenon shows that MFLM-GCN is less sensitive to parameter selection and has strong adaptability and reliability.

5. Conclusions

This article proposes MFLM-GCN to explore latent structural nuances and multi-relationship within knowledge graphs. Leveraging multi-head attention mechanisms, we meticulously discern latent correlations amid entities, engendering a manifold of comprehensive correlation graphs. Augmenting this, a densely connected layer is integrated based on random walk graphs, unraveling deeper topological insights. Furthermore, our approach orchestrates multi-relationship fusion to amalgamate both local and global relationship data, culminating in enriched entity node representations. On the DBP15K and DWY100K datasets, MFLM-GCN's Hits@1, Hits@10, and MRR are significantly better than the baseline models, fully verifying its excellent performance and effectiveness. However, there is still room for improvement in MFLM-GCN. MFLM-GCN mainly focuses on the structural information processing of the knowledge graph, and has not yet considered the use of multi-dimensional data such as attribute information, image information, entity name semantics, etc., which may limit its performance in some complex scenarios. Our future work will focus on using more multi-dimensional information to enhance the overall effect of the entity alignment process.

CRediT authorship contribution statement

Wei Ai: Writing – original draft, Validation, Supervision, Methodology, Investigation, Conceptualization. **Yulu Liu:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Chen Wei:** Writing – review & editing, Validation, Supervision, Project administration. **Tao Meng:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Hongen Shao:** Writing – review & editing, Visualization, Software, Resources, Methodology, Data curation. **Zhixiong He:** Writing – review & editing, Validation, Investigation. **Keqin Li:** Writing – review & editing, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 69189338), Excellent Young Scholars of Hunan Province of China (Grant No. 22B0275), Hunan Natural Science Foundation General Project (Grant No. 2025JJ50380), and the Hunan Provincial Natural Science Foundation Youth Project (Grant No. 2025JJ60420).

Data availability

Data will be made available on request.

References

[1] M. Gao, J.-Y. Li, C.-H. Chen, Y. Li, J. Zhang, Z.-H. Zhan, Enhanced multi-task learning and knowledge graph-based recommender system, IEEE Trans. Knowl. Data Eng. (2023).

[2] Z. Li, L. Chen, Y. Jian, H. Wang, Y. Zhao, M. Zhang, K. Xiao, Y. Zhang, H. Deng, X. Hou, Aggregation or separation? Adaptive embedding message passing for knowledge graph completion, Inform. Sci. 691 (2025) 121639.

[3] Z. Li, W. Huang, X. Gong, X. Luo, K. Xiao, H. Deng, M. Zhang, Y. Zhang, Decoupled semantic graph neural network for knowledge graph embedding, Neurocomputing 611 (2025) 128614.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: International Semantic Web Conference, Springer, 2007, pp. 722–735.

[5] J. Chen, A. Wang, J. Chen, Y. Xiao, Z. Chu, J. Liu, J. Liang, W. Wang, CN-probase: a data-driven approach for large-scale Chinese taxonomy construction, in: 2019 IEEE 35th International Conference on Data Engineering, ICDE, IEEE, 2019, pp. 1706–1709.

[6] D.B. Lenat, CYC: A large-scale investment in knowledge infrastructure, Commun. ACM 38 (11) (1995) 33–38.

[7] X. Zhang, R. Zhang, J. Chen, J. Kim, Y. Mao, Semi-supervised entity alignment with global alignment and local information aggregation, IEEE Trans. Knowl. Data Eng. 35 (10) (2023) 10464–10477.

[8] T. Le, N. Le, B. Le, Knowledge graph embedding by relational rotation and complex convolution for link prediction, Expert Syst. Appl. 214 (2023) 119122.

[9] H. Cui, T. Peng, F. Xiao, J. Han, R. Han, L. Liu, Incorporating anticipation embedding into reinforcement learning framework for multi-hop knowledge graph question answering, Inform. Sci. 619 (2023) 745–761.

[10] W. Cai, W. Ma, L. Wei, Y. Jiang, Semi-supervised entity alignment via relation-based adaptive neighborhood matching, IEEE Trans. Knowl. Data Eng. (2022).

[11] Y. Zhang, Y. Li, X. Wei, Y. Yang, L. Liu, Y.L. Murphey, Graph matching for knowledge graph alignment using edge-coloring propagation, Pattern Recognit. (2023) 109851.

[12] Y. Zhang, X. Luo, J. Hu, M. Zhang, K. Xiao, Z. Li, Graph structure prefix injection transformer for multi-modal entity alignment, Inf. Process. Manage. 62 (3) (2025) 104048.

[13] Z. Sun, W. Hu, C. Wang, Y. Wang, Y. Qu, Revisiting embedding-based entity alignment: a robust and adaptive method, IEEE Trans. Knowl. Data Eng. (2022).

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Adv. Neural Inf. Process. Syst. 26 (2013).

[15] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, 2014.

[16] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, 2015.

[17] X. Zheng, B. Zhou, M. Li, Y.G. Wang, J. Gao, MathNet: Haar-like wavelet multiresolution analysis for graph representation learning, Knowl.-Based Syst. 273 (2023) 110609.

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

[19] H.-W. Yang, Y. Zou, P. Shi, W. Lu, J. Lin, X. Sun, Aligning cross-lingual entities with multi-aspect information, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, 2019, pp. 4431–4441.

[20] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, D. Zhao, Relation-aware entity alignment for heterogeneous knowledge graphs, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence, 2019.

[21] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, B. Yin, Hierarchical graph convolution network for traffic forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 151–159.

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

[23] S. Pei, L. Yu, R. Hoehndorf, X. Zhang, Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference, in: The World Wide Web Conference, 2019, pp. 3130–3136.

[24] H. Zhu, R. Xie, Z. Liu, M. Sun, Iterative entity alignment via joint knowledge embeddings, in: IJCAI, vol. 17, 2017, pp. 4258–4264.

[25] Z. Sun, W. Hu, Q. Zhang, Y. Qu, Bootstrapping entity alignment with knowledge graph embedding, in: IJCAI, vol. 18, 2018.

[26] Z. Sun, W. Hu, C. Li, Cross-lingual entity alignment via joint attribute-preserving embedding, in: The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16, Springer, 2017, pp. 628–644.

[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Adv. Neural Inf. Process. Syst. 26 (2013).

[28] B.D. Trisedya, J. Qi, R. Zhang, Entity alignment between knowledge graphs using attribute embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 297–304.

[29] J. Hou, C. Luo, F. Qin, Y. Shao, X. Chen, FuS-GCN: Efficient B-rep based graph convolutional networks for 3D-CAD model classification and retrieval, Adv. Eng. Informatics 56 (2023) 102008.

[30] L. Liu, J. Chang, P. Zhang, H. Qiao, S. Xiong, SASG-GCN: self-attention similarity guided graph convolutional network for multi-type lower-grade glioma classification, IEEE J. Biomed. Heal. Informatics (2023).

[31] P. Zhang, J. Chen, C. Che, L. Zhang, B. Jin, Y. Zhu, IEA-GNN: Anchor-aware graph neural network fused with information entropy for node classification and link prediction, Inform. Sci. 634 (2023) 665–676.

[32] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2016.

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

[34] Z. Liu, Y. Cao, L. Pan, J. Li, T.-S. Chua, Exploring and evaluating attributes, values, and structures for entity alignment, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 6355–6364.

[35] J.D.M.-W.C. Kenton, L.K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.

[36] Y. Wu, X. Liu, Y. Feng, Z. Wang, D. Zhao, Neighborhood matching network for entity alignment, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6477–6487.

[37] Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, Y. Qu, Knowledge graph alignment network with gated multi-hop neighborhood aggregation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 222–229.

[38] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, R. Luo, Heterogeneous graph contrastive learning for recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 544–552.

[39] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15, Springer, 2018, pp. 593–607.

[40] K. Liang, Y. Liu, S. Zhou, W. Tu, Y. Wen, X. Yang, X. Dong, X. Liu, Knowledge graph contrastive learning based on relation-symmetrical structure, IEEE Trans. Knowl. Data Eng. (2023).

[41] S. Vashishth, S. Sanyal, V. Nitin, P. Talukdar, Composition-based multi-relational graph convolutional networks, in: International Conference on Learning Representations, 2019.

[42] Y. Fang, X. Li, R. Ye, X. Tan, P. Zhao, M. Wang, Relation-aware graph convolutional networks for multi-relational network alignment, ACM Trans. Intell. Syst. Technol. 14 (2) (2023) 1–23.

[43] X. Liu, K. Zhang, Y. Liu, E. Chen, Z. Huang, L. Yue, J. Yan, RHGN: Relation-gated heterogeneous graph network for entity alignment in knowledge graphs, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 8683–8696.

[44] J. Yang, W. Zhou, L. Wei, J. Lin, J. Han, S. Hu, RE-GCN: relation enhanced graph convolutional network for entity alignment in heterogeneous knowledge graphs, in: Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part II 25, Springer, 2020, pp. 432–447.

[45] Y. Zhu, H. Liu, Z. Wu, Y. Du, Relation-aware neighborhood matching model for entity alignment, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 4749–4756.

[46] D. Yu, Y. Yang, R. Zhang, Y. Wu, Generalized multi-relational graph convolution network, in: Proceedings of the 37th International Conference on Machine Learning, 2020.

[47] B. Zhu, T. Bao, R. Han, H. Cui, J. Han, L. Liu, T. Peng, An effective knowledge graph entity alignment model based on multiple information, Neural Netw. 162 (2023) 83–98.

[48] X. Shi, B. Li, L. Chen, C. Yang, Bi-neighborhood graph neural network for cross-lingual entity alignment, Knowl.-Based Syst. (2023) 110841.

[49] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.