

A Systematic Evaluation of Graph Neural Network Based Entity Alignment

Guanwen Ding, Zhichun Wang*

School of Artificial Intelligence, Beijing Normal University, Engineering Research Center of Intelligent Technology and Educational Application,
Ministry of Education, Beijing 100875, China

Keywords: Entity alignment; Graph neural network; Message passing; Skip connection; Evaluation.

Citation: Ding G.W., Wang Z.C.: A Systematic Evaluation of Graph Neural Network Based Entity Alignment. Data Intelligence, 2026, Vol. 8, Art. No.: 2026r18. DOI: <https://doi.org/10.3724/2096-7004.di.2025.0168>

ABSTRACT

Entity Alignment (EA) aims to identify equivalent entities across different Knowledge Graphs (KGs), enabling knowledge fusion and integration. In recent years, Graph Neural Networks (GNNs) have emerged as a powerful paradigm for EA by leveraging structural information in KGs. However, most existing studies emphasize novel message passing mechanisms while overlooking other crucial GNN design components. This paper presents a comprehensive and systematic evaluation of GNN-based EA methods, focusing on three key aspects: message passing strategies, the number of GNN layers, and the construction of final entity representations. We benchmark a diverse set of GNN models originally developed for tasks such as node classification and knowledge graph completion, and we assess their adaptability to the EA task. Additionally, we explore the effectiveness of skip connection techniques, activation functions, and relational information integration. Our experiments, conducted on standard EA benchmarks including DBP15K and SRPRS, reveal several counterintuitive findings: (1) message passing is indispensable for EA; (2) many node classification GNNs are highly competitive for EA; (3) one or two GNN layers generally achieve optimal performance; and (4) activation functions have minimal impact, while skip connections significantly enhance results. This study provides a principled framework and empirical foundation for designing more effective GNN-based EA models. Code and datasets are publicly available at <https://github.com/kg-bnu/GNN-EA>.

*Corresponding author: Zhichun Wang (E-mail: zcwang@bnu.edu.cn, ORCID: 0000-0001-8797-7065).

1. INTRODUCTION

Knowledge graphs (KGs) serve as a repository for structured knowledge, capturing entities, relations, and attributes in the form of triples. However, the majority of current KGs are constructed manually or semi-automatically, which inherently leads to sparsity and incompleteness in knowledge data [1], resulting in a dearth of information regarding entities or relationships within the KG. Moreover, distinct domain-specific KGs may harbor complementary knowledge. For instance, in the domain-specific KGs of healthcare, biology, and chemistry, there may be overlapping entities, yet each only documents knowledge pertinent to its own domain. Consequently, to fully leverage the complementary information across various KGs and construct a more unified and comprehensive KG, the need for knowledge graph fusion emerges. As the cornerstone of knowledge graph fusion, Entity Alignment (EA) endeavors to identify and match entities across different KGs. Figure 1 illustrates an example of EA. Entity alignment within KGs can facilitate the integration of KGs, enhance data coverage and accuracy, and thereby bolster the performance of knowledge graph-based application systems. When applied to crossdomain KGs, EA can foster cross-domain knowledge integration, achieve effective amalgamation of knowledge from various domains, and offer solutions to cross-domain challenges, such as healthcare [2] and cross-platform product matching [3]. Additionally, richer and more complete KGs can better support information retrieval [4] and question answering [5] systems, improving the precision and relevance of user query results.

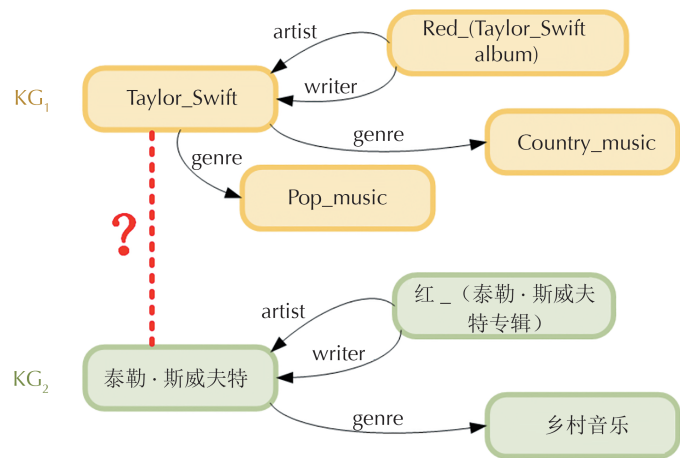


Figure 1. An example of Entity Alignment.

Graph neural networks, as a potent tool for processing graph data, can be applied to many downstream tasks of KGs, including entity alignment, node classification, and knowledge graph completion (KGC), offering innovative approaches to address challenges related to KGs. However, in the task of EA, the majority of GNN-based approaches focus on proposing new message passing (MP) methods, overlooking the impact of other configurations, and there is a scarcity of comparative studies. Previous surveys on

embedding-based entity alignment approaches [6-10] did not delve into the key features of graph neural networks. Shu et al. [11] dissected a GNN-based knowledge graph entity alignment framework into sub-components and investigated the impact of each configuration, primarily focusing on three aspects: structural embedding, alignment training, and alignment inference. Shu et al. mainly concentrated on the effects of alignment training and inference. However, for GNN encoder, they conducted experiments only on vanilla GCN and GAT, and only examined the impact of two common skip connection methods. Furthermore, apart from GNN encoder, different models exhibit varied designs in other configurations (e.g., negative sampling, loss function, distance metric, alignment strategy, bootstrap, CSLS, rerank), complicating parallel comparisons.

Therefore, to enrich the theoretical foundation of knowledge graph entity alignment and to explore the potential and limitations of graph neural networks in handling large-scale and complex graph-structured data, we conduct surveys and experiments on GNN-based EA approaches. We summarize our main contributions as follows:

- We investigate how the design of the message passing method, the number of GNN layers, and the construction of the final entity representations affect the performance of EA.
- We evaluate the performance of some message passing methods originally proposed for node classification and knowledge graph completion when adapted to EA, and also test three new skip connection methods beyond highway gates and concatenation.
- We provide a clear and focused framework for studying the GNN encoder in GNN-based EA, enabling parallel comparisons of the models' design. We release codes and datasets of this work at <https://github.com/kg-bnu/GNN-EA>.

The remainder of this paper is organized as follows. Section 2 formally defines the entity alignment task and introduces the notations used throughout the study. Section 3 presents the overall framework of GNN-based EA, detailing each component including initialization strategies, GNN encoder designs, model training, alignment prediction, and additional performance-enhancing techniques. Section 4 describes the experimental setup, including datasets, evaluation metrics, implementation details, and comprehensive analyses addressing key research questions. Finally, Section 5 concludes the paper with a summary of findings and directions for future research.

2. TASK DEFINITION

A knowledge graph can be mathematically represented as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{T}_{\mathcal{R}}, \mathcal{T}_{\mathcal{A}}\}$, where $\mathcal{E}$ denotes the set of entities, $\mathcal{R}$ represents the set of relations, and $\mathcal{A}$ signifies the set of attributes. The set of fact triples, $\mathcal{T}_{\mathcal{R}} = \{(h, r, t) \mid h, t \in \mathcal{E}; r \in \mathcal{R}\}$, indicates the presence of a relation r from the head entity h to the tail entity t . The set of attribute triples, $\mathcal{T}_{\mathcal{A}} = \{(e, a, v) \mid e \in \mathcal{E}, a \in \mathcal{A}, v \in \mathcal{V}\}$, signifies that attribute a of entity e possesses the value v , with $\mathcal{V}$ being the domain of all possible attribute values.

The process of knowledge graph entity alignment involves, given two knowledge graphs $KG_1 = \{\mathcal{E}_1, \mathcal{R}_1, \mathcal{T}_1\}$ and $KG_2 = \{\mathcal{E}_2, \mathcal{R}_2, \mathcal{T}_2\}$, identifying the set of entity correspondences $S = \{(e_i, e_j) \mid e_i \equiv e_j, e_i \in \mathcal{E}_1, e_j \in \mathcal{E}_2\}$. Here, the notation $e_i \equiv e_j$ implies that entities e_i and e_j correspond to the same real-world item or concept.

3. GNN-BASED EA FRAMEWORK

We summarized the framework of GNN-based EA in Figure 2. The GNN-based EA Framework is composed of three principal components: initialization, GNN encoder, and entity alignment. For knowledge graphs, entity embeddings are first initialized, followed by the application of graph neural encoder employing different message passing methods to aggregate neighborhood information and update the entity embeddings. These updated entity embeddings are then utilized for entity alignment. We will elucidate each component sequentially and also discuss additional designs that can enhance performance.

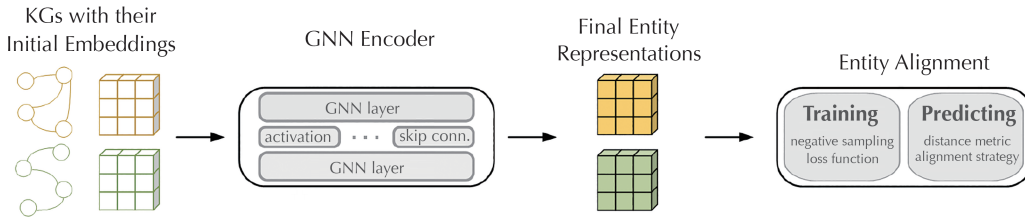


Figure 2. GNN-based Entity Alignment framework.

3.1 Initialization

There are two predominant methods for entity initialization, random initialization and name initialization:

Random Initialization. Random initialization primarily leverages the topological information of the graph for entity alignment. Representative approaches include GCN-Align [12], KECG [13], AliNet [14], MRAEA [15], RREA [16], and Dual-AMN [17].

Name Initialization. Name initialization integrates semantic information derived from topological information. HGCN [18], RDGCN [19] and RAGA [20] constructed entity name dictionaries for DBP15K [21] using GloVe^① after translating all entity names into English via Google Translate. SelfKG [22] employs LaBSE [23] (a multilingual pre-trained language model trained on 109 different languages) to construct entity name dictionaries for DBP15K and DWY100K [24].

^① huggingface.co/google-bert/bert-base-uncased.

3.2 GNN encoder

Graph Neural Networks (GNNs) are a class of deep learning models designed to operate on graph-structured data. They are particularly effective in capturing the relational structure and node features in graphs, making them suitable for a variety of tasks such as node classification, link prediction, and graph classification. In the context of Entity Alignment (EA), GNNs can be served as encoders to generate entity representations that capture both the structural and semantic information from the graph.

We categorize the design of the encoder into the following three aspects: the message passing method (single-layer GNN), the number of GNN layers, and the construction of the final entity representations (including activation functions and skip connection methods). Next, we will introduce each of these three aspects in detail.

3.2.1 Message passing methods

In the previous works on GNN-based EA, the focus has primarily been on proposing different message passing methods. A GNN can be considered as performing iterative message passing to exchange vector messages between nodes in the graph. The message passing methods can be generalized as follows:

$$\mathbf{h}_i^{(l+1)} = \text{UPDATE}^{(l)}(\mathbf{h}_i^{(l)}, \mathbf{m}_{\mathcal{N}(i)}^{(l)}) \quad (1)$$

where UPDATE is an arbitrary differentiable function, and $\mathbf{m}_{\mathcal{N}(i)}$ is the message aggregated from the neighborhood of node i , i.e. $\mathcal{N}(i)$

$$\mathbf{m}_{\mathcal{N}(i)}^{(l)} = \text{AGG}_{j \in \mathcal{N}(i)}^{(l)} \phi^{(l)}(\mathbf{h}_j^{(l)}, \mathbf{h}_i^{(l)}, \mathbf{e}_{ij}^{(l)}) \quad (2)$$

AGG is a differentiable, permutation invariant function (e.g sum, mean or max); ϕ is a function that generates messages from each node, $\mathbf{e}_{ij}$ denotes edge features from node j to node i .

Here, it is worth mentioning that we not only consider the message passing methods proposed for EA, but we include methods originally proposed for node classification and knowledge graph completion. Some representative GNN-based message passing methods are summarized as follows.

GCN-Align [12] pioneers the application of graph convolutional neural networks (GCNs) to knowledge graph entity alignment. It employs GCNs to learn entity structure embeddings and entity graph attribute embeddings, subsequently combining them, weighted by their importance, to form the final entity embedding. Additionally, GCN-Align reengineers the adjacency matrix to incorporate more relational and structural information. GCN-Align [12] leverages vanilla GCN to encode both structural and attribute features:

$$[\mathbf{H}_s^{(l+1)}; \mathbf{H}_a^{(l+1)}] = \sigma(\hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} [\mathbf{H}_s^{(l)} \mathbf{W}_s^{(l)}; \mathbf{H}_a^{(l)} \mathbf{W}_a^{(l)}]) \quad (3)$$

GraphSAGE [25] is a node classification model that, instead of training a unique embedding vector for each node, learns a set of aggregator functions to synthesize features from a node's neighborhood.

This enables the model to generalize to previously unseen nodes by employing the learned aggregation functions to generate embeddings based on local neighborhood information:

$$\begin{aligned} \mathbf{h}_{\mathcal{N}(i)}^{(l+1)} &= \text{AGGREGATE}_l(\{\mathbf{h}_j^{(l)}, \forall j \in \mathcal{N}(i)\}) \\ \mathbf{h}_i^{(l+1)} &= \sigma(\mathbf{W}^{(l+1)} \cdot \text{CONCAT}(\mathbf{h}_i^{(l)}, \mathbf{h}_{\mathcal{N}(i)}^{(l+1)})) \end{aligned} \quad (4)$$

GraphSAGE discusses three distinct architectural choices for aggregators: average aggregators, LSTM [26] aggregators, and pooling aggregators. These respectively compute the average of elements, utilize the LSTM architecture, and apply element-wise maxima post fully-connected layer processing to achieve neighborhood aggregation.

KECG [13] introduces a semi-supervised entity alignment approach grounded in the intersection graph model and joint knowledge embedding model. It extends the graph attention network (GAT) by constraining the weight matrix to a diagonal matrix, mitigating the adverse effects of heterogeneous knowledge graphs and yielding attention-enhanced entity embeddings. KECG employs a nearest-neighbor sampling strategy for negative samples, adopts TransE [27] as the knowledge embedding model, and sets the final objective function as the sum of the distancebased alignment objective and the TransE objective, thereby considering both entity alignment and semantic information. KECG utilizes an extended graph attention network (GAT) [28] with projection constraints (restricting the weight matrix to a diagonal matrix) to embed the knowledge graph:

$$\mathbf{H}^{(l+1)} = \sigma(\mathbf{A}^{(l)} \mathbf{H}^{(l)} \mathbf{W}^{(l)}) \quad (5)$$

$$a_{ij}^{(l)} = \text{softmax}(c_{ij}^{(l)}) = \frac{\exp(c_{ij}^{(l)})}{\sum_{e_k \in \mathcal{N}_{e_i} \cup \{e_j\}} \exp(c_{ik}^{(l)})} \quad (6)$$

$$c_{ij}^{(l)} = \text{LeakyReLU}(\mathbf{q}^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \oplus \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}]) \quad (7)$$

AGNN [29] is a node classification model that integrates attention mechanisms into graph neural networks, dynamically adjusting the influence of neighboring nodes on a target node. It simplifies each fully-connected layer to a scalar parameter, minimizing the number of parameters and enhancing efficiency for tasks with scarce labeled data:

$$\mathbf{H}^{(l+1)} = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \mathbf{P}_{ij}^{(l)} \mathbf{H}_j^{(l)} \quad (8)$$

where $\mathbf{P}_i^{(l)} = \text{softmax}([\beta^{(l)} \cos(\mathbf{H}_i^{(l)}, \mathbf{H}_j^{(l)})]_{j \in \mathcal{N}(i) \cup \{i\}})$, $\cos(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} / \|\mathbf{x}\| \|\mathbf{y}\|$ for the L2 paradigm. Thus, the fully connected layers are reduced to one scalar $\beta^{(l)}$ for each layer. It can be further obtained:

$$\begin{aligned} \mathbf{P}_{ij}^{(l)} &= (1/C) e^{\beta^{(l)} \cos(\mathbf{H}_i^{(l)}, \mathbf{H}_j^{(l)})} \\ C &= \sum_{j \in \mathcal{N}(i) \cup \{i\}} e^{\beta^{(l)} \cos(\mathbf{H}_i^{(l)}, \mathbf{H}_j^{(l)})} \end{aligned} \quad (9)$$

CompGCN [30] is applied to node classification, graph classification, and knowledge graph complementation (link prediction). CompGCN employs a composition operation over each edge in the neighborhood of a central node to generate messages, utilizing subtraction, multiplication, and circular-correlation operations:

$$\mathbf{e}_o = \phi(\mathbf{e}_s, \mathbf{e}_r) \mathbf{m}_{\mathcal{N}(i)}^{(l)} = \sum_{(j, r) \in \mathcal{N}(i)} \mathbf{W}_{\lambda(r)}^{(l)} \phi(\mathbf{h}_u^{(l)}, \mathbf{h}_r^{(l)}) \mathbf{h}_i^{(l+1)} = \sigma(\mathbf{m}_{\mathcal{N}(i)}^{(l)} + \mathbf{W}_0^{(l)} \mathbf{h}_i^{(l)}) \mathbf{h}_i^{(l+1)} = \mathbf{W}_r^{(l)} \mathbf{h}_r^{(l)} \quad (10)$$

VR-GCN [31] introduces relation representations in each layer of convolutional computation, enabling entities to discern whether neighboring entities are from the head or tail entity when obtaining neighboring representations, thus being influenced by the translation bias introduced by relation representations. VR-GCN [31] integrates TransE's translation operations into the message passing updates:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\left(\frac{1}{d_i} \left(\sum_{r \in \mathcal{N}_t} \sum_{t \in \mathcal{N}_t'} (\mathbf{h}_t^{(l)} - \mathbf{h}_r^{(l)}) + \sum_{r \in \mathcal{N}_h} \sum_{h \in \mathcal{N}_h'} (\mathbf{h}_h^{(l)} + \mathbf{h}_r^{(l)}) \right) + \mathbf{h}_i^{(l)} \right) \mathbf{W}^{(l)} \right) \quad (11)$$

RREA [16] employs relational reflection transformation to derive specific relational embeddings for each entity:

$$\begin{aligned} \mathbf{m}_{\mathcal{N}(i)}^{(l)} &= \sum_{(j, r) \in \mathcal{N}(i)} \alpha_{ijr}^{(l)} \mathbf{M}_r \mathbf{h}_j^{(l)} \\ \mathbf{h}_{e_i}^{(l+1)} &= \sigma(\mathbf{m}_{\mathcal{N}(i)}^{(l)}) \end{aligned} \quad (12)$$

where $\mathbf{M}_k$ is an orthogonal matrix, the relational reflection matrix of r_k ; σ is the ReLU activation function. $\alpha_{ijk}^{(l)}$ is the attention weight coefficient of $\mathbf{M}_k \mathbf{h}_j^{(l)}$, which is computed as:

$$\begin{aligned} \beta_{ijr}^{(l)} &= \mathbf{v}^\top \left[\mathbf{h}_i^{(l)} \parallel \mathbf{M}_r \mathbf{h}_j^{(l)} \parallel \mathbf{h}_r \right] \\ \alpha_{ijr}^{(l)} &= \frac{\exp(\beta_{ijr}^{(l)})}{\sum_{(j, k) \in \mathcal{N}(i)} \exp(\beta_{ijk}^{(l)})} \end{aligned} \quad (13)$$

Dual-AMN [17] enhances RREA by significantly reducing computational complexity through the combination of intelligent modeling with intra-graph and cross-graph information. It proposes a normalized hard sample mining loss to improve training efficiency and model performance. Dual-AMN [17] incorporates relational projection operations in the message passing of GNNs:

$$\begin{aligned} \mathbf{m}_{\mathcal{N}(i)}^{(l)} &= \sum_{(j, r) \in \mathcal{N}(i)} \alpha_{ijr}^{(l)} (\mathbf{h}_j^{(l)} - 2\mathbf{h}_r^\top \mathbf{h}_j^{(l)} \mathbf{h}_r) \\ \mathbf{h}_{e_i}^{(l+1)} &= \sigma(\mathbf{m}_{\mathcal{N}(i)}^{(l)}) \\ \alpha_{ijk}^{(l)} &= \frac{\exp(\mathbf{v}^\top \mathbf{h}_r)}{\sum_{(j, r') \in \mathcal{N}(i)} \exp(\mathbf{v}^\top \mathbf{h}_{r'})} \end{aligned} \quad (14)$$

3.2.2 The number of GNN layers

GNN layers can be stacked in multiple layers. Deeper GNNs can uncover intricate relationships between nodes and subgraphs that are not evident from a few layers alone. In practice, most GNN-based EA approaches would stack multiple GNN layers to learn more complex representations that capture both local and global structural information.

3.2.3 Construction of the final entity representations

The construction of final entity representations includes two techniques: activation functions and skip connection methods. Most approaches would employ an activation function between layers to introduce non-linearity. Common activation functions include ELU (Exponential Linear Unit), ReLU (Rectified Linear Unit), Tanh (Hyperbolic Tangent), and Sigmoid:

(1) ELU is designed to address the vanishing gradient problem. It has a smooth transition for negative inputs, which helps in keeping the gradients flowing even for negative values. The parameter α controls the value to which the ELU saturates for negative inputs:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases} \quad (15)$$

(2) ReLU is one of the most widely used activation functions in deep learning. It is computationally efficient and helps in mitigating the vanishing gradient problem:

$$f(x) = \max(0, x) \quad (16)$$

(3) Tanh is a scaled and shifted version of the sigmoid function. It maps the input to a value between -1 and 1 , which can help in centering the data. This centering can lead to faster convergence during training:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (17)$$

(4) Sigmoid function maps the input to a value between 0 and 1 . It is commonly used in binary classification tasks, particularly in the output layer:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (18)$$

Skip connections are architectural components that allow the flow of information to skip one or more layers in a neural network. They are particularly useful in deep networks to ease the training process and improve the flow of gradients. Two common skip connection methods include:

(1) Highway Gates (highway): Inspired by the highway networks [32] in computer vision, it uses a gating mechanism to control the flow of information. The output of a layer is computed as a combination of the transformed input and the input itself, weighted by a learned gate. For an L-Layer GNN encoder,

each layer updates using formulas (19) (20), and ultimately, the output of the last layer is taken as the final representations as formula (21):

$$T(\mathbf{h}_i^{(l)}) = \sigma(\mathbf{h}_i^{(l)} \mathbf{W}_T^{(l)} + \mathbf{b}_T^{(l)}) \quad (19)$$

$$\mathbf{h}_i^{(l+1)} = T(\mathbf{h}_i^{(l)}) \mathbf{h}_i^{(l+1)} + (1 - T(\mathbf{h}_i^{(l)})) \mathbf{h}_i^{(l)} \quad (20)$$

$$\mathbf{h}_i^{\text{highway}} = \mathbf{h}_i^{(L)} \quad (21)$$

(2) Concatenation (concat-all): This method concatenates the initial embeddings and all outputs of each GNN layer. It is straightforward and effective in preserving the information from earlier layers:

$$\mathbf{h}_i^{\text{concat-all}} = \mathbf{h}_i^{(0)} \parallel \mathbf{h}_i^{(1)} \parallel \dots \parallel \mathbf{h}_i^{(L)} \quad (22)$$

Meanwhile, some approaches do not employ skip connections (No-skip_conn.). For an L-layer GNN encoder, the output from the L-th layer is directly utilized as the final entity representations:

$$\mathbf{h}_i^{\text{No-skip_conn.}} = \mathbf{h}_i^{(L)} \quad (23)$$

Figure 3 illustrates how these methods construct the final output representation of an entity, using a two-layer GNN as an example. Ultimately, the GNN encoder produces final entity representations for alignment.

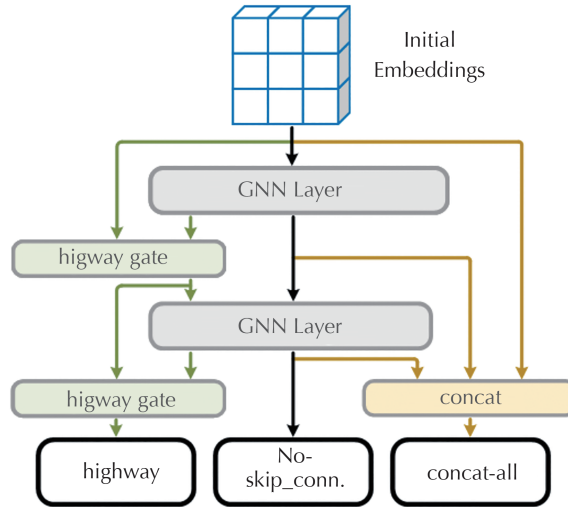


Figure 3. Skip connection methods.

3.3 Model Training

Model training involves two important aspects: loss function and negative sampling strategy.

3.3.1 Loss Function

Loss functions are critical in training models to distinguish effectively between similar and dissimilar entities. The triplet loss is the most commonly used, involving an anchor, a positive sample, and a negative sample. The objective is to minimize the distance between the anchor and the positive sample while maximizing the distance to the negative sample, ensuring a margin between these distances. This helps the model learn to better represent entities in the embedding space, bringing similar entities closer and pushing dissimilar ones apart.

3.3.2 Negative Sampling

Negative sampling involves selecting negative samples to contrast with positive samples during training. Two prevalent strategies are uniform sampling, which randomly selects negative samples from all entities, and nearest neighbor sampling, which selects from the closest entities to the positive sample. This process is essential for optimizing the loss function, as it aids the model in understanding the boundaries between different entities.

3.4 Alignment Prediction

Alignment Prediction also involves two important aspects: distance metric and alignment strategy.

3.4.1 Distance Metric

Distance metric assesses the similarity between entity embeddings from different knowledge graphs. After training to unify the embeddings into a shared vector space, the distance between corresponding entity vectors is calculated to determine their similarity based on their embedding representations. Commonly used distance metrics include L1 distance (Manhattan), L2 distance (Euclidean), and cosine similarity.

3.4.2 Alignment Strategy

Alignment strategy determines the method for finding corresponding entities in another KG. Based on the distance metric, there are three general alignment strategies: nearest neighbor search, stable matching, and iterative matching. Nearest neighbor search calculates distances between an entity and all target entities, selecting the closest as its alignment. Stable matching conducts a bidirectional search across both graphs, aligning only pairs that are closest to each other. Iterative matching continuously updates the distance between two entities based on the mapping properties of their connected relations, gradually refining the alignment.

Additionally, if the design of the entity alignment component is modified, this EA framework can be easily adapted to a GNN-based node classification or knowledge graph completion framework, as the designs are similar except for the entity alignment component.

3.5 Other designs

Beyond the basic components discussed, some approaches incorporate additional techniques to enhance performance:

3.5.1 Bootstrapping

Bootstrapping has been introduced as a strategic technique to mitigate the issue of limited labeled data in the context of entity alignment within knowledge graphs. Specifically, BootEA, NAEA, TransEdge, and MRAEA utilize bootstrapping to iteratively produce semi-supervised data. Bootstrapping enables these approaches to incrementally develop semi-supervised structured data, thereby enhancing the learning process in environments where extensive labeled datasets are not available.

3.5.2 Cosine Similarity-based Least Squares (CSLS)

CSLS was developed to tackle the hubness problem inherent in high-dimensional spaces, which is particularly relevant in the alignment of entities across different knowledge graphs. AliNet, MRAEA, RREA, and Dual-AMN have integrated CSLS into their approaches. CSLS employs a least-squares optimization technique to align word embeddings from disparate semantic spaces, effectively minimizing the discrepancy in their cosine similarity. This technique not only improves the matching of semantically related words across different languages or domains but also augments the performance of cross-lingual or domain adaptation tasks.

3.5.3 Reranking

Reranking has been proposed as an enhancement to the process of capturing entity similarity more accurately. Reranking refines the initial ranking by recalibrating it based on the bidirectional similarity of entities, thereby facilitating the identification of entity pairs that exhibit high mutual similarity. This technique is instrumental in improving the precision and robustness of entity alignment by further optimizing the similarity matching process. ERAlign [33], a model focused on semantic matching, incorporates reranking to fine-tune its alignment outcomes, thereby achieving more accurate and reliable entity alignments.

4. EXPERIMENTS

4.1 Models

We utilize the message passing mechanisms from GCN-Align [12], GraphSAGE [25], KECG [13], AGNN [29], VR-GCN [31], CompGCN [30], RREA [16], and Dual-AMN [17] as the foundational and comparative models for our experiments. We employ its average aggregator for Graph-SAGE. Among these models, GraphSAGE and AGNN were originally proposed for knowledge graph node classification tasks, while CompGCN was designed for node classification, graph classification, and knowledge graph completion (link prediction). The remaining models were specifically developed for knowledge graph entity alignment tasks.

4.2 Datasets

In this study, we utilize DBP15K [21] and SRPRS [34] as our datasets. DBP15K is the most frequently used dataset for knowledge graph entity alignment, comprising three subsets derived from DBpedia [35]: Chinese-English (DBP15K_{ZH-EN}), Japanese-English (DBP15K_{JA-EN}), and French-English (DBP15K_{FR-EN}), each containing 15,000 pre-aligned entity pairs. SRPRS is another commonly used dataset, consisting of English-French (SRPRS_{EN-FR}) and English-German (SRPRS_{EN-DE}) subsets, each also containing 15,000 pre-aligned entity pairs. Compared to DBP15K, SRPRS exhibits a sparser degree distribution and fewer relations, making it more representative of real-world knowledge graphs and allowing it to better reflect the model's reasoning ability in scenarios with limited information. Table 1 presents statistical information for both datasets, including the sizes of their entity sets, relation sets, fact triples, and the number of pre-aligned entity pairs.

Table 1. Dataset statistics.

Dataset	Lang.	$ \mathcal{E} $	$ \mathcal{R} $	$ \mathcal{T}_{\mathcal{R}} $	$ S $
DBP15K _{ZH-EN}	ZH	19388	1701	70414	15000
	EN	19572	1323	95142	
DBP15K _{JA-EN}	JA	19814	1299	77214	15000
	EN	19780	1153	93484	
DBP15K _{FR-EN}	FR	19661	903	105998	15000
	EN	19993	1208	115722	
SRPRS _{EN-FR}	EN	15000	221	36508	15000
	FR	15000	177	33532	
SRPRS _{EN-DE}	EN	15000	222	38363	15000
	DE	15000	120	37377	

Due to the limited number of pre-aligned entity pairs, which typically constitute only a small fraction of all entity pairs because of the high cost of manual alignment, existing approaches usually reserve 30% of the pre-aligned entity pairs as training data. Following the configurations of GCN-Align [12], KECG [13], HGCN [18], and other works, we also randomly divide 30% of the pre-aligned entity pairs into the training set and 70% into the test set. To ensure consistency in the input information during experiments, we also add reverse edges and self-loops to the fact triples of all datasets.

For the entity name initialization, we utilize the translated embedding dictionary provided by HGCN [18] for DBP15K, which includes a 300-dimensional embedding for each entity. For SRPRS, since no existing translated embedding dictionary is available, we construct an entity name embedding dictionary with the

same 300 dimensions per entity. We employ BERT [36][Ⓢ] to generate entity name embeddings. Specifically, we first translate the entity names in French and German into English using Google Translate, and then use BERT to obtain the word embeddings for each word in the entity names. By averaging the word embeddings, we convert them into 300-dimensional vectors through a fully connected layer. In this study, we fix the random seed and generate a fully connected layer with fixed weights initialized by Xavier and fixed biases set to zero to project the 768-dimensional entity name embeddings onto 300 dimensions.

4.3 Evaluation metrics and hyperparameters

In this paper, we adopt the hit rate (Hits@K) and Mean Reciprocal Rank (MRR) as evaluation metrics, consistent with several previous approaches [11-13, 18]. Hits@K represents the proportion of correctly aligned entities ranked within the top K, with Hits@1 specifically indicating the accuracy rate. MRR is the average of the inverted ranks of correctly aligned entities. For both Hits@N and MRR, higher values indicate better model performance.

In our experiments, we fix e_1 and seek entities aligned with e_1 in $\mathcal{E}_2$, although it is also feasible to fix e_2 and find entities aligned with e_2 in $\mathcal{E}_1$. We present the results for Hits@1, Hits@10, and MRR, focusing on the dataset from the left to the right graph.

For other configurations and hyperparameters, we select nearest neighbor sampling as the negative sampling strategy and triplet loss as the loss function. We select L2 distance as the distance metric and nearest neighbor search as the alignment strategy. The dimensions of all entity embeddings, relation embeddings, and GNN layers are set to 300. We adopt an early stopping mechanism, with the maximum number of epochs set to 1,000, and terminate training once the best validation performance is achieved for each model. Additionally, we sample 25 negative examples for each positive example every 10 iterations using nearest-neighbor sampling. The learning rate is set to 0.005, employing the Adam [37] optimizer.

4.4 Results and analysis

RQ1: The Design of the Message Passing Method

RQ1.1: Does Message Passing Really Help EA?

To ascertain whether a graph neural network in entity alignment (EA) can be substituted with a multilayer perceptron (MLP), we conducted experiments employing MLP layers on the DBP15K_{ZH-EN} and SRPRS_{EN-FR} datasets. These experiments were conducted with randomly initialized entity embeddings, using two MLP layers with ReLU activation and without skip connection.

Upon reviewing the experimental results presented in Table 2, it is evident that the MLP does not perform similarly to methods like GCN-Align and KECG. The Hits@1 for MLP is 0.0001%, akin to the performance when using directly initialized random embeddings (No-MP), suggesting that MLP fails to

[Ⓢ] huggingface.co/google-bert/bert-base-uncased.

Table 2. The impact of message passing method.

MP Method	DBP15K _{ZH-EN}			SRPRS _{EN-FR}		
	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
No-MP	0.0001	0.0011	0.001	0.0000	0.0010	0.001
MLP	0.0001	0.0010	0.001	0.0001	0.0014	0.001
FiLM [38]	0.0001	0.0010	0.001	0.0142	0.0613	0.031
FA [39]	0.0001	0.0011	0.001	0.0000	0.0010	0.001
SG [40]	0.0095	0.0253	0.016	0.0165	0.0492	0.029
SSG [41]	0.0257	0.0502	0.036	0.0286	0.0598	0.040
EG [42]	0.0363	0.0904	0.057	0.0555	0.1226	0.080
GEN [43]	0.0461	0.1441	0.080	0.0457	0.1566	0.085
CompGCN [30]	0.0931	0.1746	0.123	0.0626	0.1123	0.080
TAG [44]	0.1057	0.2989	0.170	0.0989	0.2616	0.152
GCN2 [45]	0.1742	0.3756	0.244	0.1437	0.3166	0.202
VR-GCN [31]	0.1990	0.5707	0.318	0.0631	0.2531	0.126
GATv2 [46]	0.2043	0.5578	0.320	0.1930	0.5004	0.295
GAT [47]	0.2241	0.5865	0.344	0.2007	0.5228	0.305
AntiSymmetric [48]	0.2262	0.4061	0.292	0.1305	0.2522	0.174
GCN [49]	0.2285	0.5758	0.344	0.1855	0.4710	0.278
SuperGAT [50]	0.2371	0.6172	0.362	0.1772	0.4879	0.281
Transformer [51]	0.2595	0.5091	0.346	0.1610	0.3820	0.235
GraphSAGE [25]	0.2652	0.5224	0.354	0.1760	0.4126	0.255
ClusterGCN [52]	0.3439	0.6193	0.442	0.2294	0.4906	0.317
KECG [13]	0.3788	0.7579	0.507	0.2727	0.5787	0.373
Dual-AMN [17]	0.4277	0.7950	0.553	0.3039	0.6155	0.406
RREA [16]	0.4301	0.7963	0.555	0.3040	0.6040	0.404
APPNP [53]	0.4312	0.7452	0.537	0.2636	0.5693	0.363
GCN-Align [12]	0.4383	0.7956	0.559	0.2822	0.5913	0.383

We categorize message passing (MP) methods into three groups based on their Hits@1 values on the DBP-15K_{ZH-EN} dataset, using 5% and 30% as the thresholds.

effectively update entity representations and does not contribute to entity alignment. This outcome diverges from the findings by Li et al. [54] on knowledge graph completion tasks, indicating that **for knowledge graph entity alignment, the message passing mechanism remains crucial and cannot be simply replaced with an MLP.**

RQ1.2: Does the Type of Messaging Passing Matters?

In our analysis of message passing methods, we selected a variety of models including GCNAlign [12], KECG [13], GCN [49], GAT [47], GATv2 [46], GraphSAGE [25], AGNN [29], SG [40], SSG [41], AntiSymmetric [48], Transformer [51], APPNP [53], TAG [44], ClusterGCN [52], GEN [43], FiLM [38], GCN2 [45], SuperGAT [50], FA [39], EG [42], VRGCN [31], CompGCN [30], RREA [16] and Dual-AMN [17]. Among these, GCN-Align, KECG, VR-GCN, RREA, and Dual-AMN are specifically designed for entity alignment, while CompGCN is intended for node classification, graph classification, and knowledge graph completion. The remaining models are intended for node classification. To ensure consistency and fairness in our comparison, we excluded node attribute information. For instance, with GCNAlign, we utilized the GCN(SE) variant, focusing solely on structural information. With randomly initialized entity embeddings, two layers of the GNN, ReLU activation, and no skip connection, we conducted experiments on DBP15K_{ZH-EN} and SRPRS_{EN-FR} to assess the impact of each message passing method.

The experimental results in Table 2 reveal that out of the 24 tested message passing methods, six (SG, SSG, GEN, FiLM, FA, and EG) had Hits@1 below 5%, rendering them ineffective for entity alignment, while 18 methods with Hits@1 above 5% were considered effective. Notably, AGNN, APPNP, and ClusterGCN, originally designed for node classification, achieved over 30% Hits@1, with 40.02%, 43.12%, and 34.39% respectively, comparable to GCN-Align's 43.83% and KECG's 37.88%. Similar trends were observed on SRPRS_{EN-FR}. This suggests that **many methods developed for node classification are also viable for entity alignment**, indicating that the specific message passing mechanism may not be as critical.

RQ1.3: Does the Use of Relation Information Helps?

Regarding the utilization of relation information, VR-GCN, CompGCN, RREA, and Dual-AMN are among the message passing methods that incorporate such information. The results in Table 2 show that RREA and Dual-AMN perform better, while VR-GCN and CompGCN do not outperform methods that do not use relation information, suggesting that **the introduction of relational information can improve model performance, but it is also necessary to pay attention to dealing with the noise therein.**

RQ2: Choice of the Number of Layers of GNN

We experimented with models featuring 1, 2, 3, and 5 layers, selecting representative methods (GCN-Align, KECG, GraphSAGE, AGNN, VR-GCN, CompGCN, RREA, Dual-AMN) for message passing. Using ReLU activation and no skip connection, the outcomes are presented in Table 3.

Table 3. Results of using different number of layers in GNNs.

MP Method	#Layer	Random Initialization					Name Initialization				
		DBP15K _{ZH-EN}			SRPRS _{EN-EN}		DBP15K _{ZH-EN}			SRPRS _{EN-EN}	
		Hits@1	Hits@10	MRR	Hits@1	MRR	Hits@1	Hits@10	MRR	Hits@1	MRR
GCN-Align [12]	1	0.4197	0.7671	0.539	0.2687	0.366	0.5225	0.7826	0.615	0.7010	0.7861
	2	0.4383	0.7956	0.559	0.2822	0.5912	0.5584	0.8690	0.670	0.6624	0.8113
	3	0.3376	0.7179	0.464	0.2531	0.355	0.4599	0.8275	0.586	0.5569	0.7820
	5	0.2422	0.6097	0.364	0.2482	0.353	0.3706	0.7668	0.502	0.4365	0.7266
KECG [13]	1	0.4109	0.7647	0.530	0.2488	0.349	0.5379	0.8093	0.634	0.6955	0.8266
	2	0.3788	0.7579	0.507	0.2727	0.373	0.5032	0.8309	0.617	0.5419	0.7238
	3	0.3416	0.7241	0.469	0.2584	0.364	0.4337	0.8030	0.559	0.4245	0.6682
	5	0.2434	0.6049	0.362	0.2393	0.341	0.3205	0.7147	0.450	0.2802	0.5827
GraphSAGE [25]	1	0.3966	0.6471	0.486	0.2527	0.4978	0.6415	0.7742	0.688	0.8259	0.8830
	2	0.2652	0.5224	0.354	0.1760	0.255	0.6069	0.7500	0.657	0.8002	0.8577
	3	0.1078	0.2861	0.169	0.0780	0.132	0.5731	0.7360	0.629	0.7586	0.8293
	5	0.0064	0.0327	0.016	0.0051	0.0274	0.4637	0.6819	0.538	0.6488	0.7578
AGNN [29]	1	0.3891	0.7310	0.505	0.2498	0.345	0.5565	0.8230	0.653	0.7228	0.8410
	2	0.4002	0.7610	0.523	0.2653	0.367	0.5247	0.8425	0.637	0.6582	0.8389
	3	0.3420	0.7220	0.469	0.2343	0.337	0.4571	0.8253	0.584	0.5371	0.7898
	5	0.2477	0.6138	0.367	0.2046	0.306	0.3688	0.7650	0.500	0.4383	0.7225

Table 3. Continued.

MP Method	#Layer	Random Initialization				Name Initialization			
		DBP15K _{ZH-EN}		SRPRS _{EN-HR}		DBP15K _{ZH-EN}		SRPRS _{EN-HR}	
		Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
VR-GCN [31]	1	0.1997	0.5040	0.300	0.1256	0.3021	0.186	0.4920	0.7730
	2	0.1995	0.5707	0.318	0.0631	0.2531	0.126	0.2444	0.5284
	3	0.1470	0.5094	0.260	0.0217	0.1051	0.052	0.0936	0.3065
	5	0.0008	0.0075	0.004	0.0079	0.0472	0.025	0.0127	0.0745
CompGCN [30]	1	0.2114	0.3926	0.277	0.1342	0.2456	0.174	0.6076	0.7574
	2	0.0931	0.1746	0.123	0.0626	0.1123	0.080	0.4767	0.6898
	3	0.0310	0.0783	0.048	0.0241	0.0627	0.038	0.3197	0.5111
	5	0.0052	0.0255	0.013	0.0057	0.0380	0.018	0.1302	0.2958
RREA [16]	1	0.4041	0.7614	0.525	0.2729	0.5682	0.370	0.6038	0.8565
	2	0.4299	0.7963	0.555	0.3041	0.6041	0.404	0.5606	0.8551
	3	0.4115	0.7873	0.541	0.2458	0.5549	0.347	0.4878	0.8371
	5	0.3692	0.7646	0.502	0.2510	0.5675	0.354	0.4196	0.8027
Dual-AMN [17]	1	0.3934	0.7586	0.517	0.2633	0.5650	0.364	0.6006	0.8501
	2	0.4275	0.7950	0.553	0.3039	0.6155	0.406	0.6671	0.9006
	3	0.3417	0.7278	0.470	0.2468	0.5564	0.347	0.6226	0.8927
	5	0.3750	0.7693	0.509	0.2574	0.5710	0.360	0.6196	0.8794

The experimental results indicate that on both DBP15K_{ZH-EN} and SRPRS_{EN-FR}, the eight models achieve optimal performance with 1 or 2 layers, with a significant decrease from 2 to 3 layers. For instance, GraphSAGE's Hits@1 dropped from 39.66% at 1 layer to 26.53% at 2 layers to 0.64% at 5 layers. This aligns with Li et al. [55]'s findings on semi-supervised node classification, suggesting that excessive layers can lead to over-smoothing also in knowledge graph entity alignment. When using name initialization with KECG, GraphSAGE, AGNN, VRGCN, CompGCN, and RREA, Hits@1 also deteriorated with more layers, implying that deeper networks might compromise the original textual features and reduce entity feature expressiveness. We posit that the number of GNN layers correlates with the neighborhood size aggregated by each node; excessive depth might introduce excessive noise, degrading performance. Thus, **we recommend using GNNs with 1 to 2 layers.**

RQ3: Construction of the Final Representations for Alignment

Following the results from subsection (Q2), we selected the two-layer 300-dimensional GCNAlign model for further experiments, with results detailed in Table 4.

Regarding the impact of activation functions, we observed that with GCN-Align and on DBP15K_{ZH-EN}, except in cases of random initialization and no skip connection, ReLU showed about a 3% increase compared to no activation function, models without activation functions performed similarly or better, especially with Sigmoid. It suggests that **activation functions have a minimal impact on performance and are not essential for GNN encoder design.** It also implies that the multilayered GNN encoder should be considered as an integrated whole, rather than as a series of layers separated by activation functions.

For skip connection methods, our findings align with Shu et al. [11], with the addition of three new methods: with random initialization and no activation functions, skip connection provided a modest boost. For DBP15K_{ZH-EN} Hits@1, concat-all and highway gates improved performance by 1.8% and 1.92%, respectively, over no skip connection. With name initialization, all skip connection methods significantly improved performance, with highway gates and concat-all showing the most significant gains of 13.92% and 12.89% on DBP15K_{ZH-EN}. This indicates that **skip connection methods balance textual and structural information from entity names and knowledge graphs**, enhancing alignment performance.

Furthermore, Shu et al. [11] conducted experiments using vanilla GCN and GAT message passing methods, concluding that "In general, the utilisation of highway gates leads to superior performance compared to concatenation." We extended this comparison to evaluate the effects of concatenation and highway gates with no activation function across eight representative message passing methods. The results are presented in Table 5. Upon analysis, it appears that **the superiority of highway gates over concatenation is not pronounced**, and the relative effectiveness of the two methods may depend on both the dataset and the specific message passing method. Therefore, in practical applications, it may still be necessary to empirically compare these two methods.

Table 4. The impact of activation function and skip connection (MP method is GCN-Align).

Activation function	Skip connection	Random Initialization					Name Initialization				
		DBP15K _{ZH-EN}			SRPRS _{EN-FR}		DBP15K _{ZH-EN}			SRPRS _{EN-FR}	
		Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1
No-activation	No-skip_conn.	0.4067	0.7764	0.532	0.2621	0.5750	0.365	0.5656	0.8744	0.677	0.6684
	concat-all	0.4247	0.7500	0.536	0.2831	0.5676	0.376	0.7048	0.8382	0.753	0.8278
	highway	0.4259	0.7649	0.540	0.2551	0.5637	0.356	0.6945	0.8625	0.755	0.8159
	concat-all(highway)	0.3697	0.7047	0.483	0.2427	0.5469	0.342	0.6674	0.8004	0.714	0.8197
	concat-0&L	0.3159	0.7105	0.447	0.2335	0.5438	0.336	0.6774	0.8177	0.727	0.8150
	residual	0.4156	0.7670	0.533	0.2511	0.5678	0.354	0.6417	0.8272	0.709	0.7950
ELU	No-skip_conn.	0.4074	0.7768	0.532	0.2630	0.5774	0.366	0.5698	0.8763	0.680	0.6811
	concat-all	0.4249	0.7457	0.535	0.2830	0.5680	0.375	0.6967	0.8257	0.743	0.8244
	highway	0.4040	0.7524	0.520	0.2546	0.5641	0.356	0.7041	0.8693	0.763	0.8164
	concat-all(highway)	0.3699	0.7044	0.483	0.2416	0.5454	0.341	0.6620	0.7938	0.709	0.8122
	concat-0&L	0.3159	0.7106	0.447	0.2332	0.5441	0.336	0.6705	0.8058	0.718	0.8111
	residual	0.4159	0.7671	0.533	0.2521	0.5672	0.355	0.6429	0.8240	0.708	0.7965

Table 4. Continued.

Activation function	Skip connection	Random Initialization				Name Initialization			
		DBP15K _{ZH-EN}		SRPRS _{EN-ER}		DBP15K _{ZH-EN}		SRPRS _{EN-ER}	
		Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
ReLU	No-skip_conn.	0.4382	0.7956	0.559	0.2822	0.5912	0.383	0.5584	0.8690
								0.6625	0.8113
	concat-all	0.3791	0.6668	0.479	0.2645	0.5314	0.351	0.6774	0.7975
	highway	0.4512	0.7650	0.558	0.2844	0.5667	0.377	0.6915	0.8493
								0.748	0.8247
	concat-all(highway)	0.3798	0.6886	0.485	0.2540	0.5355	0.346	0.6513	0.7786
Tanh	concat-0&L	0.2496	0.5738	0.359	0.2145	0.4882	0.304	0.6446	0.7685
	residual	0.4241	0.7565	0.537	0.2549	0.5595	0.354	0.6461	0.8124
								0.706	0.7969
	No-skip_conn.	0.4066	0.7765	0.532	0.2617	0.5774	0.365	0.5761	0.8790
								0.685	0.7005
	concat-all	0.4247	0.7500	0.536	0.2830	0.5676	0.376	0.7092	0.8412
Sigmoid	highway	0.4258	0.7650	0.540	0.2563	0.5654	0.358	0.7004	0.8696
								0.762	0.8234
	concat-all(highway)	0.3697	0.7047	0.483	0.2425	0.5470	0.342	0.6663	0.7980
	concat-0&L	0.3159	0.7105	0.447	0.2334	0.5438	0.336	0.6757	0.8119
	residual	0.4157	0.7671	0.533	0.2515	0.5676	0.354	0.6460	0.8282
								0.712	0.7957
Sigmoid	No-skip_conn.	0.3752	0.7490	0.500	0.2505	0.5519	0.349	0.4985	0.8164
								0.608	0.6668
	concat-all	0.0023	0.0076	0.005	0.0132	0.0318	0.021	0.5789	0.6947
	highway	0.2418	0.4890	0.327	0.1788	0.4089	0.256	0.6833	0.8259
								0.734	0.8256
	concat-all(highway)	0.0403	0.0631	0.050	0.0333	0.0576	0.043	0.6461	0.7672
Sigmoid	concat-0&L	0.0001	0.0011	0.001	0.0002	0.0010	0.001	0.5517	0.6543
	residual	0.3712	0.6776	0.478	0.2219	0.5041	0.314	0.6310	0.7781
								0.683	0.7912
	No-skip_conn.							0.6668	0.8278
								0.6668	0.8278
	concat-all	0.0023	0.0076	0.005	0.0132	0.0318	0.021	0.5789	0.6947
Sigmoid	highway	0.2418	0.4890	0.327	0.1788	0.4089	0.256	0.6833	0.8259
								0.734	0.8256
	concat-all(highway)	0.0403	0.0631	0.050	0.0333	0.0576	0.043	0.6461	0.7672
	concat-0&L	0.0001	0.0011	0.001	0.0002	0.0010	0.001	0.5517	0.6543
	residual	0.3712	0.6776	0.478	0.2219	0.5041	0.314	0.6310	0.7781
								0.683	0.7912

Table 5. Comparison of highway gates and concatenation.

Activation function	Skip connection	DBP15K _{ZH-EN}			SRPRS _{EN-FR}		
		Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR
GCN-Align [12]	concat-all	0.7048	0.8382	0.753	0.8278	0.8661	0.841
	highway	0.6945	0.8625	0.755	0.8159	0.8563	0.830
KECG [13]	concat-all	0.7142	0.8585	0.765	0.8301	0.8733	0.845
	highway	0.7150	0.9029	0.783	0.7850	0.8568	0.811
GraphSAGE [25]	concat-all	0.6219	0.7481	0.666	0.8161	0.8616	0.833
	highway	0.6144	0.7431	0.659	0.8057	0.8394	0.818
AGNN [29]	concat-all	0.6983	0.8347	0.748	0.8290	0.8682	0.843
	highway	0.7026	0.8736	0.765	0.7990	0.8492	0.818
VR-GCN [31]	concat-all	0.3406	0.6179	0.437	0.5605	0.7353	0.622
	highway	0.3725	0.6754	0.475	0.6350	0.8090	0.698
CompGCN [30]	concat-all	0.6121	0.7374	0.657	0.7968	0.8299	0.809
	highway	0.6021	0.7274	0.647	0.7965	0.8269	0.808
RREA [16]	concat-all	0.7073	0.8458	0.757	0.8265	0.8721	0.843
	highway	0.7177	0.9042	0.785	0.8641	0.9323	0.889
Dual-AMN [17]	concat-all	0.7141	0.8518	0.764	0.8508	0.9110	0.872
	highway	0.7241	0.8997	0.788	0.7844	0.8559	0.810

5. CONCLUSIONS

Our study provides a comprehensive analysis of the role of the message passing mechanism in knowledge graph entity alignment, highlighting its significance in both node feature extraction and alignment tasks. The findings of our research can be summarized as follows:

(1) Indispensability of Message Passing: For knowledge graph entity alignment, the message passing mechanism is deemed essential and cannot be effectively supplanted by a multilayer perceptron (MLP). This underscores the necessity of leveraging the graph structure for optimal alignment performance.

(2) Adaptability of Message Passing Methods: A variety of message passing methods, initially designed for node classification and knowledge graph completion, have proven to be readily adaptable and highly effective for entity alignment. This challenges the assumption that the specificity of these message passing methods is a critical determinant of their success in entity alignment.

(3) Enhancement and Noise Management with Relational Information: The incorporation of relational information is shown to enhance model performance. However, it also introduces the challenge of managing noise, necessitating careful consideration in model design and data processing strategies.

(4) Optimal Number of GNN Layers: Our results indicate that the use of 1 to 2 layers in graph neural networks (GNNs) generally yields superior outcomes in entity alignment tasks. This suggests a balance between a node's neighborhood size and excessive noise.

(5) Final representations of entities: The inclusion of an activation function in the GNN encoder design is found to be non-essential. Additionally, the choice between highway gates and concatenation for skip connection methods requires further empirical exploration to determine their impact on alignment accuracy and robustness.

In conclusion, our research has not only explored the potential of graph neural networks in handling large-scale, complex graph-structured data but also delineated the limitations and areas requiring further investigation. This study offers a clear framework for future research on GNN-based EA approaches, guiding the development of more effective and efficient models for knowledge graph applications.

AUTHOR CONTRIBUTIONS

Guanwen Ding: proposed the research problems; designed the research framework; performed the research; collected and analyzed the data; wrote and revised the manuscript.

Zhichun Wang: proposed the research problems; designed the research framework; wrote and revised the manuscript.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 62276026).

REFERENCES

- [1] J. Pujara, E. Augustine, and L. Getoor, "Sparsity and noise: Where knowledge graph embeddings fall short," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1751–1756. [Online]. Available: <https://aclanthology.org/D17-1184/>.
- [2] P. Yang, H. Wang, Y. Huang, S. Yang, Y. Zhang, L. Huang, Y. Zhang, G. Wang, S. Yang, L. He, and Y. Huang, "Lmkg: A large-scale and multi-source medical knowledge graph for intelligent medicine applications," *Knowledge-Based Systems*, vol. 284, pp. 111–323, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950705123010717>.
- [3] W. Liu, J. Pan, X. Zhang, X. Gong, Y. Ye, X. Zhao, X. Wang, K. Wu, H. Xiang, H. Yan, and Q. Zhang, "Cross-platform product matching based on entity alignment of knowledge graph with raea model," *World Wide*

- Web, vol. 26, no. 4, p. 22152235, Feb. 2023. [Online]. Available: <https://doi.org/10.1007/s11280-022-01134-y>.
- [4] A. Kotov and C. Zhai, "Tapping into knowledge base for concept feedback: leveraging conceptnet to improve search results for difficult queries," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, ser. WSDM '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 403–412. [Online]. Available: <https://doi.org/10.1145/2124295.2124344>.
 - [5] X. Yao and B. Van Durme, "Information extraction over structured data: Question answering with Freebase," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Toutanova and H. Wu, Eds. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 956–966. [Online]. Available: <https://aclanthology.org/P14-1090/>.
 - [6] Z. Sun, Q. Zhang, W. Hu, C. Wang, M. Chen, F. Akrami, and C. Li, "A benchmarking study of embedding-based entity alignment for knowledge graphs," *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 2326–2340, Jul. 2020. [Online]. Available: <https://doi.org/10.14778/3407790.3407828>.
 - [7] K. Zeng, C. Li, L. Hou, J. Li, and L. Feng, "A comprehensive survey of entity alignment for knowledge graphs," *AI Open*, vol. 2, pp. 1–13, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666651021000036>.
 - [8] Z. Zhang, H. Liu, J. Chen, X. Chen, B. Liu, Y. Xiang, and Y. Zheng, "An industry evaluation of embedding-based entity alignment," in *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, A. Clifton and C. Napoles, Eds. Online: International Committee on Computational Linguistics, Dec. 2020, pp. 179–189. [Online]. Available: <https://aclanthology.org/2020.coling-industry.17/>.
 - [9] X. Zhao, W. Zeng, J. Tang, W. Wang, and F. M. Suchanek, "An experimental study of state-of-the-art entity alignment approaches," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 6, pp. 2610–2625, 2022.
 - [10] B. Zhu, R. Wang, J. Wang, F. Shao, and K. Wang, "A survey: knowledge graph entity alignment research based on graph embedding," *Artificial Intelligence Review*, vol. 57, no. 9, p. 229, 2024.
 - [11] Y. Shu, J. Zhang, G. Huang, C.-H. Chi, and J. He, "Entity alignment via graph neural networks: a component-level study," *World Wide Web*, vol. 26, no. 6, pp. 4069–4092, 2023.
 - [12] Z. Wang, Q. Lv, X. Lan, and Y. Zhang, "Cross-lingual knowledge graph alignment via graph convolutional networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 349–357. [Online]. Available: <https://aclanthology.org/D18-1032/>.
 - [13] C. Li, Y. Cao, L. Hou, J. Shi, J. Li, and T.-S. Chua, "Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model," 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2723–2732. [Online]. Available: <https://aclanthology.org/D19-1274/>.
 - [14] Z. Sun, C. Wang, W. Hu, M. Chen, J. Dai, W. Zhang, and Y. Qu, "Knowledge graph alignment network with gated multi-hop neighborhood aggregation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 222–229, Apr. 2020. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/5354>.
 - [15] X. Mao, W. Wang, H. Xu, M. Lan, and Y. Wu, "MRAEA: An efficient and robust entity alignment approach for cross-lingual knowledge graph," in *Proceedings of the 13th International Conference on Web Search*

- and Data Mining, ser. WSDM '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 420–428. [Online]. Available: <https://doi.org/10.1145/3336191.3371804>.
- [16] X. Mao, W. Wang, H. Xu, Y. Wu, and M. Lan, “Relational reflection entity alignment,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1095–1104. [Online]. Available: <https://doi.org/10.1145/3340531.3412001>.
- [17] X. Mao, W. Wang, Y. Wu, and M. Lan, “Boosting the speed of entity alignment 10 ×: Dual attention matching network with normalized hard sample mining,” in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 821–832. [Online]. Available: <https://doi.org/10.1145/3442381.3449897>.
- [18] Y. Wu, X. Liu, Y. Feng, Z. Wang, and D. Zhao, “Jointly learning entity and relation representations for entity alignment,” 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)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 240–249. [Online]. Available: <https://aclanthology.org/D19-1023/>.
- [19] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao, “Relation-aware entity alignment for heterogeneous knowledge graphs,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 5278–5284. [Online]. Available: <https://doi.org/10.24963/ijcai.2019/733>.
- [20] R. Zhu, M. Ma, and P. Wang, “Raga: Relation-aware graph attention networks for global entity alignment,” in *Advances in Knowledge Discovery and Data Mining*, K. Karlapalem, H. Cheng, N. Ramakrishnan, R. K. Agrawal, P. K. Reddy, J. Srivastava, and T. Chakraborty, Eds. Cham: Springer International Publishing, 2021, pp. 501–513.
- [21] Z. Sun, W. Hu, and C. Li, “Cross-lingual entity alignment via joint attribute-preserving embedding,” in *The Semantic Web-ISWC 2017*, C. d’Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange, and J. Heflin, Eds. Cham: Springer International Publishing, 2017, pp. 628–644.
- [22] X. Liu, H. Hong, X. Wang, Z. Chen, E. Kharlamov, Y. Dong, and J. Tang, “Selfkg: Self-supervised entity alignment in knowledge graphs,” in *Proceedings of the ACM Web Conference 2022*, ser. WWW '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 860–870. [Online]. Available: <https://doi.org/10.1145/3485447.3511945>.
- [23] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang, “Language-agnostic BERT sentence embedding,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 878–891. [Online]. Available: <https://aclanthology.org/2022.acl-long.62/>.
- [24] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, “Bootstrapping entity alignment with knowledge graph embedding,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 4396–4402. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/611>.
- [25] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7e9-Paper.pdf>.
- [26] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>.

- [27] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.
- [28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- [29] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," 2018. [Online]. Available: <https://arxiv.org/abs/1803.03735>.
- [30] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BylA_C4tPr.
- [31] R. Ye, X. Li, Y. Fang, H. Zang, and M. Wang, "A vectorized relational graph convolutional network for multi-relational network alignment," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 4135–4141. [Online]. Available: <https://doi.org/10.24963/ijcai.2019/574>.
- [32] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015. [Online]. Available: <https://arxiv.org/abs/1505.00387>.
- [33] S. Yoon, S. Ko, T. Kim, S. Kang, J. Yeo, and D. Lee, "Unsupervised robust cross-lingual entity alignment via neighbor triple matching with entity and relation texts," 2025. [Online]. Available: <https://arxiv.org/abs/2407.15588>.
- [34] L. Guo, Z. Sun, and W. Hu, "Learning to exploit long-term relational dependencies in knowledge graphs," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2505–2514. [Online]. Available: <https://proceedings.mlr.press/v97/guo19c.html>.
- [35] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "Dbpedia a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015. [Online]. Available: <https://doi.org/10.3233/SW-140134>.
- [36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 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)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://aclanthology.org/N19-1423/>.
- [37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>.
- [38] M. Brockschmidt, "GNN-FiLM: Graph neural networks with feature-wise linear modulation," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1144–1152. [Online]. Available: <https://proceedings.mlr.press/v119/brockschmidt20a.html>.

- [39] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 3950–3957, May 2021. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/16514>.
- [40] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6861–6871. [Online]. Available: <https://proceedings.mlr.press/v97/wu19e.html>.
- [41] H. Zhu and P. Koniusz, "Simple spectral graph convolution," in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=CYO5T-YjWZV>.
- [42] S. A. Tailor, F. Opolka, P. Lio, and N. D. Lane, "Adaptive filters for low-latency and memory-efficient graph neural networks," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=hl9ePdHO4_s.
- [43] G. Li, C. Xiong, A. Thabet, and B. Ghanem, "Deepergcn: All you need to train deeper gcns," 2020. [Online]. Available: <https://arxiv.org/abs/2006.07739>.
- [44] J. Du, S. Zhang, G. Wu, J. M. F. Moura, and S. Kar, "Topology adaptive graph convolutional networks," 2018. [Online]. Available: <https://openreview.net/forum?id=H113pWZRb>.
- [45] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1725–1735. [Online]. Available: <https://proceedings.mlr.press/v119/chen20v.html>.
- [46] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" in *International Conference on Learning Representations*, 2022. [Online]. Available: <https://openreview.net/forum?id=F72ximsx7C1>.
- [47] P. Velickovi, G. Cucurull, A. Casanova, A. Romero, P. Li, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=rjXmpikCZ>.
- [48] A. Gravina, D. Bacciu, and C. Gallicchio, "Anti-symmetric DGN: a stable architecture for deep graph networks," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=J3Y7cgZOOS>.
- [49] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017. [Online]. Available: <https://openreview.net/forum?id=SJU4ayYgl>.
- [50] D. Kim and A. Oh, "How to find your friendly neighborhood: Graph attention design with selfsupervision," in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=Wi5KUNlqWty>.
- [51] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 1548–1554, main Track. [Online]. Available: <https://doi.org/10.24963/ijcai.2021/214>.
- [52] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 257–266. [Online]. Available: <https://doi.org/10.1145/3292500.3330925>.

- [53] J. Gasteiger, A. Bojchevski, and S. Gnnemann, “Combining neural networks with personalized pagerank for classification on graphs,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=H1gL-2A9Ym>.
- [54] J. Li, H. Shomer, J. Ding, Y. Wang, Y. Ma, N. Shah, J. Tang, and D. Yin, “Are message passing neural networks really helpful for knowledge graph completion?” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 10696–10711. [Online]. Available: <https://aclanthology.org/2023.acl-long.597/>.
- [55] Q. Li, Z. Han, and X.-M. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/11604>.

AUTHOR BIOGRAPHY



Guanwen Ding is a graduated student at the School of Artificial Intelligence, Beijing Normal University. He received his bachelor's degree from China University of Mining and Technology in 2024. His research interests include knowledge graph and language models.



Zhichun Wang is currently an associate professor at the School of Artificial Intelligence, Beijing Normal University. He received his Ph.D. degree from Tianjin University in 2010, and worked as post-doctoral researcher in Tsinghua University from 2011 to 2012. His research interests include large language models and knowledge graph.