ABSTRACT
Seeking the equivalent entities among multi-source Knowledge Graphs (KGs) is the pivotal step to KGs integration, also known as entity alignment (EA). However, most existing EA methods are inefficient and poor in scalability. A recent summary points out that some of them even require several days to deal with a dataset containing 200,000 nodes (DWY100K). We believe over-complex graph encoder and inefficient negative sampling strategy are the two main reasons. In this paper, we propose a novel KG encoder — Dual Attention Matching Network (Dual-AMN), which not only models both intra-graph and cross-graph information smartly, but also greatly reduces computational complexity. Furthermore, we propose the Normalized Hard Sample Mining Loss to smoothly select hard negative samples with reduced loss shift. The experimental results on widely used public datasets indicate that our method achieves both high accuracy and high efficiency. On DWY100K, the whole running process of our method could be finished in 1, 100 seconds, at least 10× faster than previous work. The performances of our method also outperform previous works across all datasets, where Hits@1 and MRR have been improved from 6% to 13%.

CCS CONCEPTS
• Computing methodologies → Knowledge representation and reasoning: Natural language processing; Supervised learning.

KEYWORDS
Graph Neural Networks; Knowledge Graph; Entity Alignment

1 INTRODUCTION
Typically, knowledge graphs (KGs) store real-world knowledge in the form of triples (i.e., <entity, relation, entity>), where entities are connected through relations. In recent years, many general KGs (e.g., DBpedia [1], YAGO [19]) and domain-specific KGs (e.g., Scientific [25]) have proliferated and been widely used in downstream applications, such as search engines and recommendation systems.

In practice, a KG is usually constructed from one single data source. Therefore, it is unlikely to cover the full domain. As shown in Figure 1(a), integrating KGs in the same domain but built from different languages can transfer the information from high resource language to low resource language. This in turn will facilitate downstream cross-lingual applications, especially for minority language users. Moreover, consolidating multi-domain KGs (Figure 1(b)) can supplement cross-domain information and improve the coverage, thus making KGs more complete.

Seeking the equivalent entities among multi-source KGs is the pivotal step to KGs integration, also known as entity alignment (EA). Recently, EA attracts enormous attention and progresses rapidly. Dozens of related papers have been published in recent years. In general, these methods all share one core framework: assume that equivalent entities possess similar neighboring structure, apply KG embedding methods (e.g., TransE [2] or GCN [9]) to obtain dense embeddings for each entity, then map these embeddings into a unified vector space by alignment module (e.g., Triplet loss and

Figure 1: Two examples of KGs integration. Figure (a) represents cross-lingual and Figure (b) represents cross-domain.
Contrastive Loss[6, 16]), and finally the pair-wise distance between entities determines whether they are aligned or not.

However, previous EA methods are inefficient and poor in scalability, as summarized by Zhao et al. [34] that most of them require several hours [3, 11, 22] or even days [31] on a dataset containing 200,000 nodes (i.e., DWY100K). In reality, KGs usually consist of millions of entities and relations (e.g., the full DBpedia contains 10+ billion entities, 1+ trillion triples). Such large-scale datasets impose huge challenges in the efficiency and scalability of EA methods. Obviously, the high cost in time hinders the feasibility of applying these EA methods to large-scale KGs.

We believe there are two main reasons that cause the high time complexity of these advanced methods:

1. **Over-complex graph encoder:** Since vanilla GCN is unable to model the heterogeneous relation information in KGs, many relation-aware GNN variants are proposed in EA task. However, some GNN variants are over-complex and inefficient. The running time of the vanilla GCN [27] is only 10% of those from complex encoders. Every time a complex technique is introduced, e.g., Graph Attention mechanism [26], Graph Matching Networks [12] (GMN), Joint Learning [11], the time complexity is dramatically increased. For instance, GM-Align [31] incorporates GMN and achieves decent improvements on a small dataset (DBP15K), but the performance improvement most likely is contributed from the literal information. When moving to a larger dataset (DWY100K), GM-Align needs five days to obtain the results. We believe the graph encoder still has some redundancy in design and its architecture can be further simplified to reduce time consumption.

2. **Inefficient negative sampling strategy:** Almost all existing EA methods rely on the pair-wise loss functions (e.g., TransE, Triplet loss, and Contrastive Loss). In pair-wise loss, the negative samples are constructed via uniform random sampling. In this way, the samples are usually highly redundant and have limited information. The learning process could be hampered by the low-quality negative samples, resulting in slow convergence and model degradation. To alleviate this problem, BootEA [22] proposes a Truncated Uniform Negative Sampling strategy to choose K-nearest neighbors as negative samples (i.e., hard samples). Such an intuitive and effective strategy has been widely adopted in subsequent studies [3, 11, 23]. However, ranking all neighbors to find the K-nearest is highly time-consuming and difficult to be fully parallelized on GPU. For example, Truncated Uniform Negative Sampling Strategy takes more than 25% of the whole time cost of BootEA.

Instead of trading efficiency for better performance, in this paper, we propose Dual Attention Matching Network (Dual-AMN) to capture dual relational information within a single graph and across two graphs. The Simplified Relational Attention Layer captures relational information within each KG by generating relation-specific embeddings through Relational Anisotropy Attention and Relational Projection. The Proxy Matching Attention Layer treats alignment as a special relation type and explicitly models it via proxy vectors. In addition, to tackle the inefficient sampling issue, we further propose a Normalized Hard Sample Mining Loss. First, LogSumExp operation is used to approximate Max operation to generate hard samples smoothly but efficiently. Then, to resolve the dilemma of hyper-parameter selection in LogSumExp, we introduce a loss normalization strategy adjusting the distribution of loss dynamically.

Experiment with the same hardware environment, our method could finish the whole running process in 1, 100 seconds on DWY-100K, including data loading, training, and evaluating, which is 3× faster compared to the fastest existing model (i.e., GCN-Align [27]) and only takes up 10% of advanced methods. On DBP15K with a smaller scale, our method even could obtain results in less than 40 seconds. More surprisingly, the alignment results obtained by our method have very high accuracy. The experiments show that our method beats all state-of-the-art competitors across all datasets, and the performance improvement on hits@1 and MRR ranges from 6% to 13%. The main contributions are summarized as follows:

- **Model.** We propose a novel graph encoder Dual Attention Matching Network (Dual-AMN) composed of Simplified Relational Attention Layer and Proxy Matching Attention Layer. The proposed encoder not only models both intra-graph and cross-graph relations smartly, but also greatly reduces computational complexity.

- **Training.** Instead of the inefficient sampling strategy, we propose a Normalized Hard Sample Mining Loss, where the LogSumExp operation generates hard samples efficiently and the loss normalization alleviates the dilemma of hyper-parameter selection. The new loss dramatically cuts down the sampling consumption and accelerates the convergence speed of the model.

- **Experiments.** The experimental results on widely used public datasets indicate that our method has high efficiency and accuracy. Furthermore, we design many auxiliary experiments to demonstrate the effectiveness of each component and the interpretability of the model.

### 2 TASK DEFINITION

**Definition of Knowledge Graph:** The formal definition of a KG is a directed graph \( G = (E, R, T) \) comprising three sets — entities \( E \), relations \( R \), and triples \( T \subseteq E \times R \times E \). KG stores the real-world information in the form of triples \(<\text{entity}, \text{relation}, \text{entity}>\), which describe the inherent relation between two entities. In addition, we define \( N_{e_i} \) to represent the neighbor set of entity \( e_i \) and \( R_{ij} \) represent the set of relations between \( e_i \) and \( e_j \).

**Definition of Entity Alignment:** Given two KGs \( G_1 = (E_1, R_1, T_1) \), \( G_2 = (E_2, R_2, T_2) \), and a pre-aligned entity pair set \( P = \{(u, v) | u \in E_1, v \in E_2 \} \), where \( \equiv \) denotes equivalence. EA aims to obtain more potential equivalent entity pairs based on the information of \( G_1, G_2, \) and \( P \).

### 3 RELATED WORK

As mentioned in Section 1, existing EA methods can be abstracted into one framework containing three major components:

- **Graph embedding module** is responsible for encoding entities and relations of KGs into dense embeddings.

- **Entity alignment module** aims to map the embeddings of multi-source KGs into a unified vector space via pre-aligned entity pairs.

- **Information enhancement module** is able to generate semi-supervised data or introduce additional literal information for enhancement.
TransE [2], GNN, and Hybrid are the three mainstream embedding mechanisms to efficiently capture the long-term relational dependencies. They integrate Recurrent Neural Networks (RNNs) with a skipping mechanism to enhance the expression ability of the model. However, vanilla GNN [9] is unable to encode heterogeneous relations, and TransR [13] are proposed and proven to be effective in subgraph matching tasks. Based on this hypothesis, many variants (e.g., TransH [28]) are proposed and have been proved effective in improving alignment performance.

Table 1: Categorization of some popular EA approaches.

| Method      | Embedding | Alignment   | Enhancement |
|-------------|-----------|-------------|-------------|
| MTransE [4] | TransE    | Mapping     | None        |
| GCN-Align [27] | GNN      | Margin-based | None        |
| RSNs [5]    | RSNs      | Corpus fusion | None       |
| MuGNN [3]   | Hybrid    | Margin-based | None        |
| KEGC [11]   | Hybrid    | Margin-based | None        |
| BootEA [22] | TransE    | Corpus fusion | Semi-supervised |
| NAEP [35]   | Hybrid    | Corpus fusion | Semi-supervised |
| TransEdge [23] | TransE | Semi-supervised | None        |
| MRAEA [14]  | GNN       | Margin-based | Semi-supervised |
| GM-Align [31] | GNN      | Margin-based | Entity Name |
| RDGCN [29]  | GNN       | Margin-based | Entity Name |
| HMAN [32]   | GNN       | Margin-based | Attribute   |
| HGCN [30]   | GNN       | Margin-based | Entity Name |

In this section, we categorize existing EA approaches based on their designs of these three components, as shown in Table 1.

3.1 Embedding Module

TransE [2], GNN, and Hybrid are the three mainstream embedding approaches. TransE interprets relations as the translation from head entities to tail entities and assumes that the embeddings of entities and relations follow the assumption $h + r = t$ if a triple $(h, r, t)$ holds. Based on this hypothesis, many variants (e.g., TransH [28] and TransR [13]) are proposed and proven to be effective in subsequent studies. Graph Neural Network (GNN) is famous for its strong modeling capability on the non-Euclidean structure. Different from TransE optimizing triples, GNN generates node-aware embeddings by aggregating the neighboring information of entities. However, vanilla GNN [9] is unable to encode heterogeneous relational graphs such as KGs. Thus, many subsequent studies focus on modifying GNN to fit into KG. The main direction is to use the anisotropic attention mechanism [26] to assign different weight coefficients to entities. A GNN model whose node update equation treats every edge direction equally, is considered isotropic (e.g., vanilla GCN); and a GNN model whose node update equation treats every edge direction differently, is considered anisotropic (e.g., GAT [26]). Hybrid embedding approaches combine TransE and GNN together, which aim to enhance the expression ability of the model. However, for now, the best-performing methods TransEdge [23] and MRAEA [14] are not hybrid. The hybrid-based methods do not show necessity while introducing additional complexity.

In addition to these three mainstream approaches, RSNs [5] integrates Recurrent Neural Networks (RNNs) with a skipping mechanism to efficiently capture the long-term relational dependencies within and between KGs. RSNs performs well on sparse KGs, but it is still weaker than SOTA mainstream methods.

3.2 Alignment Module

The most common alignment methods are as follows: (1) Mapping [4] uses one or two linear transformation matrices to map the embeddings of entities in different KGs into a unified vector space. This idea is inspired by the cross-lingual word embedding task [10], and the first proposed EA method [4] adopts this alignment module. (2) Corpus fusion [22] swaps the entities in the pre-aligned set and generates new triples to calibrate the embeddings into a unified space. For example, there are two triples $(e_1, r_1, e_2) \in KG_1$ and $(e_3, r_2, e_4) \in KG_2$. If $e_1 \equiv e_3$ holds, Corpus fusion adds two extra triples $(e_3, r_1, e_2)$ and $(e_1, r_2, e_4)$. This approach not only integrates two KGs into one KG but also plays the role of data augmentation. (3) Margin-based represents a series of pair-wise margin-based loss functions, such as Triplet loss [16], Contrastive loss [6], and so on. Margin-based loss functions are often combined with Siamese Neural Network in ranking tasks (e.g., cross-lingual word embedding task [10], relation classification [17], etc.). Actually, GNN-based EA methods are inspired by the Siamese Neural Network and have similar architecture, so most of them use Margin-based loss to be their alignment module.

3.3 Enhancement Module

Because manually aligning entities is expensive in practice, pre-aligned pairs are usually a small part of all entities. Therefore, existing methods usually reserve 30% or even less of the aligned pairs as training data to simulate this situation. Due to the lack of labeled data, some EA methods [22, 23] adopt bootstrapping to generate semi-supervised data iteratively. Based on the asymmetric nature of cross-KG alignment, MRAEA [14] further proposes a bi-directional iterative strategy. These data augmentation techniques have been proved effective in improving alignment performance.

In addition to structure, some methods [30, 32] propose that introducing literal information could provide a multi-aspect view for alignment models and improve accuracy. However, it should be noted that not all datasets contain literal information, especially in practical applications. For example, there are privacy risks when using User Generated Content (UGC). Compared with the literal methods, the structure-only methods are more general. Therefore, these literal methods should be compared among themselves.

4 DUAL ATTENTION MATCHING NETWORK

As mentioned in Section 1, existing graph encoders, which are over-complex in certain designs and poor in scalability, are not suitable to be applied to large-scale KG. To address these defects, we propose Dual Attention Matching Network (Dual-AMN). Figure 2 depicts that Dual-AMN is composed of two major components: Simplified Relational Attention Layer and Proxy Matching Attention Layer. The Simplified Relational Attention Layer captures relational information within each KG by generating relation-specific embeddings through Relational Anisotropy Attention and Relational Projection. By treating alignment as a special relation, our Proxy Matching Attention Layer leverages a list of proxies to explicitly capture the cross-graph information. By combining the outcomes of these two...
proposed components, our Dual-AMN not only embeds both intra- 
graph and cross-graph relations smartly, but also greatly reduces 
computational complexity. The experimental results show that the 
proposed method achieves the SOTA in both performance and effi- 
ciency. In this section, we describe the architecture of Dual-AMN 
in details.

4.1 Simplified Relational Attention Layer
Since vanilla GCN is unable to model the heterogeneous relation 
information in KGs, many relation-aware GNN variants are 
proposed in EA task. Most of them could be described by the following 
equation:

$$h_{e_i}^{l+1} = \sum_{e_j \in N_{e_i} \cup \{e_i\}} \alpha_{ij}W h_{e_j}^l$$

(1)

where $h_{e_j}^l$ represents the embedding vector of $e_j$ obtained by the $l$-th 
GNN layer, $\alpha_{ij}$ represents the weight coefficient between $e_i$ and $e_j$, 
$W$ represents the transformation matrix. Table 2 lists some popular 
GNN encoders. We summarize three findings: (1) Except GCN-
Align which first utilizes GCN in EA, all the other methods adopt 
anisotropic attention mechanism. This indicates that it is necessary 
to distinguish the importance of entities. (2) There is a tendency 
that more recent methods are not joint learning based, probably 
because joint methods are not superior in performance. For example, 
MRAEA and TransEdge outperform MuGNN, KECG, and NAEA. 
So joint learning which introduces extra computation complexity 
is not necessary. (3) We also notice that many methods constrain 
the transformation matrix $W$ of GNN layer to be diagonal or even 
remove $W$ in order to avoid performance degradation. We believe 
the main reason is that the entity embeddings are all trainable 
and the standard linear transformation may introduce too many 
parameters, causing over-fitting when updating these embeddings.

Inspired by these findings, we design a simplified relation-aware 
GNN layer.

The inputs of our model are two metrics, $H^e \in \mathbb{R}^{|E| \times d}$ represents 
the initial entity features and $H^r \in \mathbb{R}^{|R| \times d}$ represents the initial relation 
features. Both of them are randomly initialized by $He_{initializer}$ 
[7]. Similar to existing EA methods, we use anisotropic relational 
attention mechanism to aggregate the neighborhood information 
around entities. The output embedding of entity $e_i$ at the $l$-th layer is 
obtained by the following equation:

$$h_{e_i}^{l+1} = \tanh \left( \sum_{e_j \in N_{e_i}} \sum_{r \in R_{ij}} \alpha_{ijk} (h_{e_j}^l - 2h_{r_k}^l h_{r_k}^l) \right)$$

(2)

here we employ $\tanh$ as the activation function. Instead of standard 
linear transformation matrix $W$, we utilize Relational Projection op- 
eration [15]. Such operation generates relation-specific embedding 
for each entity without extra parameters. As for the calculation of

| Method             | $\alpha_{ij}$ | Joint | $W$ |
|--------------------|---------------|-------|-----|
| GCN-Align [27]     | Isotropic     | ×     | None |
| MuGNN [3]          | Anisotropy    | ✓     | Diagonal |
| KECG [11]          | Anisotropy    | ✓     | Diagonal |
| MRAEA [14]         | Anisotropy    | ✓     | None |
| NAEA [35]          | Anisotropy    | ✓     | Normal |
| HMAM [32]          | Anisotropy    | ×     | Diagonal |

Table 2: Categorization of GNN encoders in some popular 
EA approaches.
\( \alpha_{ijk} \), we adopt the meta-path [33] mechanism to assign weights:

\[
\alpha_{ijk} = \frac{\exp(\langle \beta \rangle^T h_{r_{ik}})}{\sum_{j' \in R_i} \sum_{r_{ij'} \in R_i'} \exp(\langle \beta \rangle^T h_{r_{ij'}})}
\]

where \( \langle \beta \rangle \) is an attention vector. Softmax operation selects the most critical path from all types of edges connected to the entities (i.e., meta-path), which embeds the relational anisotropy but simplifies the calculation to the greatest extent.

In previous studies [14, 24], GNN is able to expand to multi-hop neighboring level information by stacking more layers, thus to create a more global-aware representation of the graph. Following this idea, we concatenate the embeddings from different layers together to obtain the Multi-Hop Embeddings for entity \( e_i \):

\[
h_{multi} = [h_{e_i}^0, h_{e_i}^1, \ldots, h_{e_i}^{n-1}]
\]

where \( \| \) represents the concatenate operation.

### 4.2 Proxy Matching Attention Layer

So far, the GNN encoders we have discussed only focus on modeling a single KG while leaving the cross-graph information to be learned by the alignment module alone. Graph Matching Network [12] (GMN) builds a cross-graph attention mechanism to learn similarities, although they view the alignment purely as a node-to-node interaction (as illustrated in Figure 3(a)). Formally, GMN measures the difference between \( e_i \in E_1 \) and its closest neighbor in the other graph as follows:

\[
\beta_{ij} = \frac{\exp(\text{sim}(h_{e_i}, h_{e_j}))}{\sum_{e_k \in E_2} \exp(\text{sim}(h_{e_i}, h_{e_k}))}
\]

\[
h_{diff}^{e_i} = \sum_{e_j \in E_2} \beta_{ij}(h_{e_j} - h_{e_i})
\]

\( \text{sim}(\cdot) \) is a vector space similarity metric, \( h_{diff}^{e_i} \) represents the difference of \( e_i \) against all entities from \( G_2 \). Such node-to-node interaction enforces the embeddings to be learned jointly on a pair, at the cost of massive extra computation efficiency. Since attention weights are required for every pair of nodes across two graphs, this operation has a computation cost of \( O(|E_1||E_2|) \). As mentioned in Section 1, GM-Align which incorporates GMN needs several days to obtain the results on the large-scale dataset (DWY100K). Driven by similar motivation, but in our interpretation, alignment itself is nothing but a special relation type whose representation can be explicitly learned in early stage.

Inspired by the above, we propose the Proxy Matching Attention Layer. As shown in Figure 3 (b), we employ a limited set of proxy vectors to represent the cross-graph alignment relation, similar to use anchor points to present a space. If two entities are equivalent, their similarity distributions associated with these proxy vectors should also be consistent. In this way, the proposed layer is able to capture the cross-graph alignment information without computing node-to-node interaction. The interaction of the Proxy Matching Attention Layer is to calculate the similarity between all entities and limited anchors, which is similar to clustering. On large-scale KGs or dense graphs, this interaction approach can greatly reduce the computational complexity from \( O(|E_1||E_2|) \) to \( O(|E_1| + |E_2|) \).

The inputs of the Proxy Matching Attention Layer are two matrices: \( H^{multi} \in \mathbb{R}^{n \times |E_1||E_2|} \) represents the entities embeddings obtained by the Simplified Relational Attention Layer and \( Q \in \mathbb{R}^{n \times |E_1||E_2|} \) represents proxy vectors with random initialization, where \( n \) represents the number of proxy vectors. Just like GMN, the first step is to compute the similarity between each entity and all proxy vectors:

\[
\beta_{ij} = \frac{\exp(\text{cos}(h_{multi}^{e_i}, q_j))}{\sum_{k \in S_p} \exp(\text{cos}(h_{e_i}, q_k))}
\]

\( S_p \) represents the set of proxy vectors. Here we use the cosine metric to measure the similarity between embeddings. Then, the cross-graph embedding for entity \( e_i \) can be computed as:

\[
h_{e_i}^p = \sum_{j \in S_p} \beta_{ij}(h_{multi}^{e_i} - q_j)
\]

\( h_{e_i}^p \) intuitively describes the difference between \( h_{multi}^{e_i} \) and all proxy vectors. Finally, we employ a gate mechanism [18] to combine \( h_{multi}^{e_i} \) and \( h_{e_i}^p \), controlling the information flow between single graph and multiple graphs:

\[
\eta_e = \text{sigmoid}(Mh_{e_i}^p + b)
\]

\[
h_{e_i}^{final} = \eta_{e_i} h_{e_i}^p + (1 - \eta_{e_i}) \cdot h_{e_i}^p
\]

\( M \) and \( b \) are the gate weight matrix and gate bias vector.

### 5 NORMALIZED HARD SAMPLE MINING

Typically, in KGs, only a small portion of cross-graph entity pairs are aligned. So negative sampling is crucial to EA methods. However, the most common approach which selects the K-nearest neighbors, spends a lot of time on candidate ranking in each epoch. In this section, we propose a Normalized Hard Sampling Mining strategy, which is efficient and reduces loss shift.
5.1 Smooth Hard Sample Mining

Both TransE-based and GNN-based EA methods rely on the pairwise loss functions to optimize the similarities between samples. TransE-based methods use TransE loss to encode KGs:

\[
L = \sum_{(h,r,t) \in T} \left[ y + \|h + r - t\|^2 - \|h' + r' - t'\|^2 \right]_+
\]

(11)

GNN-based methods use Triplet loss to map the embeddings from two KGs into a unified space:

\[
L = \sum_{(e_i, e_j) \in P} \left[ y + \text{sim}(e_i, e_j) - \text{sim}(e_i', e_j') \right]_+
\]

(12)

where \(y\) represents a fixed margin, \([x]_+\) represents the operation \(\text{Max}(0, x)\), \(x'\) represents the negative sample of \(x\).

Initially, the negative samples in pair-wise loss are generated through uniform random sampling, but this kind of samples is highly redundant and comprises too many easy uninformative samples. Training with such low-quality negative samples may significantly degrade the model’s learning capability and slow down the convergence. A simple but effective strategy is to select the K-nearest neighbors around the positive sample to be negative samples. This is also known as Hard Sample Mining. BootEA proposes the Truncated Uniform Negative Sampling (TUNS) based on this strategy and reports that it could significantly reduce the number of training epochs and improve performance. Most of the subsequent works follow this approach, such as KECG [11], MuGNN [3], TransEdge [21], and etc. However, faster convergence cannot shorten the overall training time. Because it has to spend massive time in candidate ranking for the next epoch, and this process is difficult to be fully parallelized on GPU.

In the field of deep metric learning, some studies [17, 20] propose to use the LogSumExp operation to smoothly generate hard negative samples:

\[
L = \log \left[ 1 + \sum_{i \in P} \sum_{j \in N} \exp(\lambda(y + s_i - s_j)) \right]
\]

(13)

where \(P\) represents the positive sample of the anchor set and \(N\) represents the negative sample set. \(\lambda\) is a scale factor. If \(\lambda \to \infty:\)

\[
L = \lim_{\lambda \to \infty} \frac{1}{\lambda} \log \left[ 1 + \sum_{i \in P} \sum_{j \in N} \exp(\lambda(y + s_i - s_j)) \right] = \text{Max}(y + s_i - s_j)_+
\]

(14)

LogSumExp is approximate to TUNS with \(K = 1\). When \(\lambda\) is set to an appropriate value. LogSumExp could replace the K-nearest sampling strategy to generate high-quality negative samples, but with better computational efficiency (because this process could be fully parallelized on GPU). More interestingly, when \(\lambda = 1\), the loss function is equivalent to Softmax with Cross-Entropy loss. This also indicates that the classification losses and the pair-wise losses are essentially two sides of the same coin.

5.2 Loss Normalization

Both TUNS and LogSumExp face the same dilemma of how to select the proper value for their hyper-parameters. In TUNS, the hyper-parameter is the number of nearest neighbors \(K\). A small \(K\) will lead to slow convergence in the initial training process, while an overly large \(K\) makes the negative samples too “easy.” In LogSumExp operation, the hyper-parameter is the scalar factor \(\lambda\). As illustrated in Table 3, if \(\lambda\) is set too large, the weights of samples are greatly affected by the random disturbance at the beginning of training. For example, when \(\lambda = 100\), these five pair losses are closer to each other while their corresponding weights vary a lot. In such a case, the model would tend to only focus on a few samples, slowing down the convergence. On the other hand, if \(\lambda\) is too small, it would be difficult for the model to pick up hard samples in the later stage, which causes model degradation. For example, when \(\lambda = 5\), though \(l_1\) is seven times larger than \(l_2\), the weights difference is small.

Inspired by batch normalization [8] which reduces the internal covariate shift, we propose to use a normalization step that fixes the mean and variance of samples. This is done by the following steps:

\[
l(h(e_i, e_j, e_j')) = \frac{l_o(e_i, e_j, e_j') - \mu(e_i, e_j)}{\sqrt{\sigma^2(e_i, e_j) - \epsilon}}
\]

(16)

where \(l_o(e_i, e_j, e_j')\) represents the original loss of the triple \((e_i, e_j, e_j')\), \(\mu\) and \(\sigma^2\) represent the mean and the variance of original loss, which are computed by:

\[
\mu(e_i, e_j) = \frac{1}{|E_2|} \sum_{e_j' \in E_2} l_o(e_i, e_j, e_j')
\]

(18)

\[
\sigma^2(e_i, e_j) = \frac{1}{|E_2|} \sum_{e_j' \in E_2} \left[ l_o(e_i, e_j, e_j') - \mu(e_i, e_j) \right]^2
\]

(19)

The calculation process of \(l_n(e_i, e_j, e_j')\) is similar to \(l_o(e_i, e_j, e_j')\).
During training, we choose L2 distance as the metric to measure the similarity between entities:

\[
\text{sim}(e_i, e_j) = \|h_i^{\text{final}} - h_j^{\text{final}}\|^2
\]

(20)

During testing, in order to address the hubness problem in high-dimensional space, CSLS [10] is set to be the distance metric. Note that in training, \(\sigma\) and \(\mu\) won’t participate in gradient calculation and backpropagation. This is because our loss normalization is designed to change the weights of the samples, not the gradient direction. If \(\sigma\) and \(\mu\) are updated in the backpropagation step, our loss will fail to converge.

6 EXPERIMENTS

We use the Keras framework for developing our approach. Our experiments are conducted on a workstation with a GeForce GTX TITAN X GPU and 128GB memory, which is consistent with the summary [34]. The code is now available on GitHub ¹.

6.1 Datasets

To fairly and comprehensively verify the effectiveness, robustness and scalability of our model, we construct experiments on three widely used public datasets:

(1) **DBP15K** [21]: This dataset consists of three cross-lingual subsets constructed from DBpedia: English-French (DBP\textsubscript{EN-FR}), English-Chinese (DBP\textsubscript{EN-ZH}), and English-Japanese (DBP\textsubscript{EN-JA}). Each subset contains 15,000 pre-aligned entity pairs for training and testing. As an early dataset, DBP15K is widely used but has some defects: small scale and dense links. These defects prompt more datasets to be proposed.

(2) **DWY100K** [22]: This dataset comprises two mono-lingual subsets, each containing 100,000 pre-aligned entity pairs and nearly one million triples. DWY\textsubscript{DBP-WD} represents the subset extracted from DBpedia and Wikidata, and DWY\textsubscript{DBP-YAGO} represents DBpedia and YAGO. DWY100K, as the largest dataset of the three, brings challenges to space and time complexity.

(3) **SRPRS** [5]: Compared with the real-world KGs, the above two datasets are too dense, and the degree distribution is quite different from the real. Thus, Guo et al. [5] propose a sparse dataset, including two cross-lingual subsets (SRPRS\textsubscript{FR-EN} and SRPRS\textsubscript{DE-EN}) and two mono-lingual subsets (SRPRS\textsubscript{DBP-WD} and SRPRS\textsubscript{DBP-YAGO}). Same with DBP15K, each subset of SRPRS contains 15,000 pre-aligned entity pairs for training and testing. This dataset challenges the modeling ability of EA approaches when facing limited information.

The statistics of these datasets are listed in Table 4. Consistent with previous studies, we randomly split 30% of the pre-aligned entity pairs for training and developing, while the remaining 70% for testing.

### Table 4: Statistical data of DBP15K, DWY100K and SRPRS.

| Datasets | | | |
|----------|-----------------|----------|----------|
|          | DBpedia         | YAGO     | Wikipedia |
| DWY\textsubscript{DBP-YG} | 100,000 | 302 | 428,952 |
| DWY\textsubscript{DBP-WD} | 100,000 | 31 | 502,565 |
| DWY\textsubscript{DBP-YAGO} | 100,000 | 330 | 463,294 |
| DBP\textsubscript{ZH-EN} | 19,388 | 1,701 | 70,414 |
| DBP\textsubscript{JA-EN} | 19,814 | 1,299 | 77,214 |
| DBP\textsubscript{FR-EN} | 19,780 | 1,153 | 93,484 |
| SRPRS\textsubscript{FR-EN} | 15,000 | 177 | 33,532 |
| SRPRS\textsubscript{DE-EN} | 15,000 | 222 | 38,363 |
| SRPRS\textsubscript{DBP-WD} | 15,000 | 223 | 33,748 |
| SRPRS\textsubscript{DBP-YAGO} | 15,000 | 30 | 36,569 |
| SRPRS\textsubscript{DBP-Wikipedia} | 15,000 | 253 | 38,421 |
|          | 15,000 | 144 | 40,159 |

¹https://github.com/MaoXinn/Dual-AMN

6.2 Baselines

As mentioned in Section 3, many studies adopt enhancement modules. For instance, GM-Align and RDGCN propose to introduce literal information to provide a multi-aspect view. The introduction of additional information leads to unfair comparisons between methods. Thus, existing EA methods will be compared separately according to the enhancement category:

(1) **Basic**: This kind of method only uses the original structure information (i.e., triples) in the dataset and does not introduce any extra enhancement module: MTransE [4], GCN-Align [27], RSNs [5], MuGNN [3], KEGC [11].

(2) **Semi-supervised**: These methods adopt bootstrapping to generate semi-supervised structure data: BootEA [22], NAEA [35], TransEdge [23], and MRAEA [14].

(3) **Literal**: To obtain a multi-aspect view, literal methods use literal information (e.g., entity name) as input features: GM-Align [31], RDGCN [29], HMAN [32], HGCN [30].

To make a fair comparison against above three types of methods, our model also has three corresponding versions: (1) Dual-AMN is the basic version without any enhancement module, as described in Section 4. (2) Dual-AMN (Semi) introduces the bi-directional iterative strategy proposed by MRAEA to generate semi-supervised data. (3) Dual-AMN (Lit) adopts a simple strategy to utilize literal information. For \(e_i \in KG_1\) and \(e_j \in KG_2\), we use Dual-AMN (Semi) to obtain the structural similarity \(s_{ij}\). Then, using the cross-lingual word embedding to calculate the literal similarity \(l_{ij}\). Finally, the entities are ranked according to \(l_{ij} + s_{ij}\).

6.3 Experimental Settings

**Metrics.** Following convention, we use \(\text{Hits@k}\) and \(\text{Mean Reciprocal Rank (MRR)}\) as our evaluation metrics. The \(\text{Hits@k}\) score is calculated by measuring the proportion of correctly aligned pairs in the top-k. In particular, \(\text{Hits@1}\) represents accuracy. In order to be convincing, the reported performance is the average of five independent training runs. **Hyper-parameters.** For all dataset, we use a same config: The dimensionality for embeddings \(d = 100\); depth of GNN \(l = 2\);
We compare the performances within each category.

Table 6: Experimental results on mono-lingual datasets. Because of the memory limitation, NAEA cannot work on DWY 100.

| Method     | DWY| DBP | DBP | SRPRS| SRPRS |
|------------|----|-----|-----|------|-------|
|            | @1| @10| @10| @10 | @10   |
| MTransE    | 0.209 | 0.512 | 0.310 | 0.250 | 0.360 |
| GCN-Align  | 0.434 | 0.762 | 0.550 | 0.427 | 0.540 |
| MuGNN      | 0.494 | 0.844 | 0.611 | 0.501 | 0.621 |
| KEGC       | 0.477 | 0.835 | 0.598 | 0.489 | 0.610 |
| RSNs       | 0.508 | 0.745 | 0.591 | 0.507 | 0.737 |
| Dual-AMN   | 0.731 | 0.923 | 0.799 | 0.726 | 0.927 |
| BootEA     | 0.629 | 0.847 | 0.703 | 0.622 | 0.853 |
| NAEA       | 0.650 | 0.867 | 0.720 | 0.641 | 0.872 |
| TransEdge  | 0.735 | 0.919 | 0.801 | 0.719 | 0.932 |
| MRAEA      | 0.757 | 0.930 | 0.827 | 0.758 | 0.934 |
| Dual-AMN   | 0.808 | 0.940 | 0.857 | 0.801 | 0.949 |

6.4 Main Experiments

In Table 5 and Table 6, we report the performances of all methods on cross-lingual datasets and mono-lingual datasets, respectively. We compare the performances within each category.

Dual-AMN vs. Basic Methods. Our method consistently achieves the best performance across all datasets. On the small-scale dense dataset (DBP15K), Dual-AMN outperforms other methods by at least 20% in terms of both Hits@1 and MRR. On the large-scale dense dataset (DWY100K), the performances are increased by more than 15% compared to previous SOTA. Experimental results show that the designs of Dual-AMN effectively captures the rich structural information of these two datasets. By cutting down the number of triples, SRPRS challenges EA methods’ ability to model sparse KGs. It is not surprising to see that the performances of all methods drop significantly compared to the results on dense datasets. RSNs outperforms the previous SOTA on this dataset, which could be credited to the long-term relational paths it captures. But our Dual-AMN still achieves the best performance, exceeding RSNs by at least 10% on Hits@1 and MRR. All these experimental results demonstrate the effectiveness of Dual-AMN in capturing the structural information.

Dual-AMN vs. Semi-supervised Methods. Benefiting from the semi-supervised strategy to generate more labeled data for the next training round, the overall performances of the semi-supervised methods surpass the basic methods. Compared with previous SOTA methods, our method outperforms at least 5% on Hits@1. Compared to its basic version, the semi-supervised strategy greatly improves the performances on DBP15K and DWY100K. On SRPRS, although the semi-supervised strategy still has some benefit, the number of proxy vectors $n = 64$; margin $y = 1$; new mean and new variance of normalized loss are $\tau = 10$ and $\lambda = 30$; batch size is 1024; dropout rate is set to 30%. RMSprop is adopted to optimize the model with learning rate set to 0.005.
enhancement is reduced to 2% − 3%. We believe the reason for the smaller improvement is because the sparse nature of SRPRS makes its structure information insufficient to generate high-quality semi-supervised data. Overall, the semi-supervised strategy performs well on dense datasets, while its improvement is marginal in sparse datasets.

**Dual-AMN vs. Literal Methods.** According to Zhao et al. [34], because the entity names between mono-lingual KGs are almost identical, the edit distance algorithm could achieve the ground-truth performance. Therefore, the literal methods only experiment on cross-lingual datasets.

By combining with cross-lingual embeddings, the performances of Dual-AMN are further improved and surpass the previous SOTA methods across all datasets. From observing Table 5, we found that the performances of the literal methods vary significantly according to language pairs, which is completely different from the structure-only methods. On DBP15K, the introduction of literal information increases Hits@1 by 6%, 9%, and 11%, which indicates that French is the most similar language to English, while Chinese is the most different. Besides, due to the lack of structural information, the literal information is more critical on SRPRS. Literal information improves the performances by 30% on Hits@1.

It must be admitted that our way of utilizing literal information is too simple and crude. Compared with other methods, performance improvement mainly comes from better structural embeddings. How to better integrate literal information is our future work.

**Efficiency Analysis.** Better performance is just the cherry on the cake. Dual-AMN’s trump card is superior efficiency. Table 7 reports the overall time costs of existing EA methods on each dataset, including data loading, pre-processing, training, and evaluating. All results are obtained by directly running the source code provided by the authors. And hyper-parameters are set to be the same as reported in their original papers. Certainly, implement details such as learning rate, batch size, and pre-processing might influence the time costs. However, we believe that these experimental results still reflect the overall efficiency of EA methods.

| Method            | DBP15K  | SRPRS  | DWY100K |
|-------------------|---------|--------|---------|
| MTransE [4]       | 6.467   | 3.355  | 70.085  |
| GCN-Align [27]    | 103     | 87     | 3.212   |
| RSNs [5]          | 7.539   | 2.602  | 28.516  |
| MuGNN [3]         | 3.156   | 2.215  | 47.735  |
| KECG [11]         | 3.724   | 1.800  | 125.386 |
| Dual-AMN          | 35      | 27     | 1.948   |
| BootEA [22]       | 4,661   | 2,659  | 64,471  |
| NAEA [35]         | 19,115  | 11,746 | -       |
| TransEdge [23]    | 3,629   | 1,210  | 20,839  |
| MRAEA [14]        | 3,894   | 1,248  | 23,275  |
| Dual-AMN(Semi)    | 85      | 79     | 3,169   |
| GM-Align [31]     | 26,328  | 13,032 | 459,715 |
| RDGCN [29]        | 6,711   | 886    | -       |
| HMAN [32]         | 5,455   | 4,424  | 31,895  |
| HGCN [30]         | 11,275  | 2,504  | 60,005  |
| Dual-AMN(Lit) [24]| 101     | 96     | 3,257   |

**Table 7: Time costs of EA methods (seconds).**

Obviously, the efficiency of Dual-AMN far exceeds competitors. The time costs of complex EA methods are tens or even hundreds of times more than that of Dual-AMN. Even compared with the fastest baseline (i.e., GCN-Align), the speed of Dual-AMN is 3× faster, while the Hits@1 outperforms more than 20%. Comparing Dual-AMN and Dual-AMN (Semi), semi-supervised strategy increases the time consumption about three times. Due to the simple combining strategy, Dual-AMN (Lit) hardly increases the time consumption.

In particular, the large-scale dense dataset (DWY100K) poses a severe challenge to the space and time complexity of all EA methods. Due to the limitation of GPU memory, MuGNN, KECG, and HMAN have to be run on CPU, resulting in massive time costs. GM-Align is the least efficient method, because it uses GMN and requires a complicated pre-processing. We fail to obtain results for NAEA and RDGCN in our experiment environment because they require extremely high memory space. Benefit from the simplification of the encoder architecture and the Normalized Hard Sample Mining Loss, our model could fully utilize the GPU to obtain high-accuracy results efficiently. Even using the semi-supervised strategy for data augmentation, the proposed method still could obtain results within an hour.

In summary, the high efficiency of Dual-AMN makes the entity alignment application on large-scale KGs possible.

### 6.5 Ablation Experiment

**To demonstrate the effectiveness of each design in architecture and loss function, we construct two ablation experiments on DBP15K.**

**Ablation Experiment of architecture.** Dual-AMN adopts the following four components to capture multi-aspect information existing in KGs: (1) *Relational Attention* mechanism (RA) finds the critical path around entities. (2) *Relational Projection* operation (RP) generates the relation-specific embedding for entities. (3) *Multi-Hop Embeddings* (MHE) creates a more global-aware representation of the KGs. (4) *Proxy Attention Matching Layer* (PAM) captures the cross-graph information. Table 8 reports the performances with Mean5@stds, after removing these components from Dual-AMN. Among all these components, MHE has the greatest impact on performance. Without MHE, the performance is degraded by at least 6% on Hits@1. Only stacking GNN layers cannot fully capture the global information, it is necessary to concatenate the output embeddings of each layer explicitly. Besides, the remaining three components also show the necessity as our expectation. On average, adopting these technologies improves performance by 2% to
3%. By adopting these new designs, Dual-AMN further breaks the ceiling of EA accuracy.

Ablation Experiment of Loss. Besides architecture, the Normalized Hard Sample Mining Loss is also one of our main contributions. To verify its effectiveness, we compare it with several common loss functions. The results are visualized in Fig 4. Compared with the other three, the proposed loss could make the model converge faster and achieve the best performance. The Normalized Uniform Negative Sampling Strategy also has a similar decent performance. However, as we have mentioned, this sampling strategy requires massive time consumption. Since most of the negative samples are redundant, the Triplet loss has the worst efficiency of all loss functions. In our experiments, the Triplet loss function usually needs thousands of epochs to converge, and the performance is lower than the proposed loss about 4%. The performance of Softmax with Cross-Entropy is stronger than Triplet loss, but there is obviously a performance gap with Normalized Hard Sample Mining Loss. These experimental results show that the proposed loss function significantly increases the convergence speed without losing any accuracy.

6.6 Relation Interpretability
In addition to the performance and speed advantages, our model also has a certain degree of interpretability. Because the weights of adjacent entities are determined by the relations between them, thus these weights can reflect the importance of different relations to some extent. The importance of each relation is obtained by the following equation:

\[ I_r = v^T h_r \quad (21) \]

We train the model on the DWY of relations and output the importance \( I_r \) of relations. After clustering the relations according to \( I_r \), we obtain the Table 9. From the observation, we summarize an interesting phenomenon. The relations with high importance (i.e., meta-path) are usually able to identify the entity from another. For example, if holding an Person entity and Parent relation, we can reduce the potential options down to a small space. However, President relation does not have this ability. A country can have many presidents, so its importance becomes extremely low. Of course, this is inseparable from the characteristics of the DBP dataset, which contains a large number of celebrities, especially the president, prime minister, and so on. Therefore, such kinds of relations become unimportant in this dataset.

6.7 Degree Analysis
The main experiments show that the performances of all EA methods on sparse datasets are much lower than that of standard datasets. In order to further explore the correlation between model performance and dataset density, we design an experiment on DBP–EN. Figure 5 shows the Hits@1 of the three variants on different levels of entity degrees. We observe a strong correlation between performance and degree. As the degree increasing, the model performance improves significantly. For Dual-AMN, the Hits@1 of the entities with one neighbor is only 20%. The introduction of semi-supervised strategy improves the overall performance of the model, but it has a limited effect on those entities with extremely sparse local structures. In sparse graphs, it is difficult to make correct inferences only based on limited structural information. On the other hand, Dual-AMN has much higher performance when the degree value is small, which proves that the incorporation of literal information effectively improves the accuracy of these sparse entities. However, this strategy cannot work on the datasets without literal information. Therefore, how to better represent these sparse entities without extra information is a key point of future work.
7 CONCLUSION

Over complex graph encoders and inefficient negative sampling strategies lead to the general inefficiency of existing EA methods, resulting in difficulty for applying on large-scale KGs. In this paper, we propose a novel KG encoder Dual Attention Matching Network (Dual-AMN), which not only models both intra-graph and cross-graph relations smartly but also greatly reduces computational complexity. To replace the inefficient sampling strategy, we propose Normalized Hard Sample Mining Loss to cut down the sampling consumption and accelerate the convergence speed. These two modifications enable the proposed model to achieve the SOTA performance while the speed is several times than other EA methods. The main experiments indicate that our method outperforms competitors across all datasets and metrics. Furthermore, we design auxiliary experiments to demonstrate the effectiveness of each component and the interpretability of the model.

REFERENCES

[1] Sören Auers, Christian Bizer, Georgi Kobalzov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nexus for a Web of Open Data. In The Semantic Web. 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007 (Lecture Notes in Computer Science, Vol. 4825) Karl Aberer, Key-Sun Choi, Natasha Fridman, Noy, Dean Aleman, Kyung-II Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riechiro Mizoguchi, Guss Schreiber, and Philippe Cudré-Mauroux (Eds.). Springer, 722–735. https://doi.org/10.1007/978-3-540-76286-0_52

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Yann LeCun. 2013. Translating Embeddings for Modeling Multi-relational Data. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, Christopher J. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 2787–2795. http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data

[3] Yixin Cao, Zhiyuan Liu, Junruijuan Li, Zhiyuan Liu, Lui T., and Tat-Seng Chua. 2019. Multi-Channel Graph Neural Network for Entity Alignment. In Proceedings of the 57th Conference on Association for Computational Linguistics, ACL 2019. Florence, Italy, July 28—August 2, 2019, Volume 1: Long Papers. 1492–1501. https://doi.org/10.18653/v1/p19-1149

[4] Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. 2020. MRAEA: An Efficient and Robust Entity Alignment Approach for Cross-lingual Knowledge Graph. In WSDM ’20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3–7, 2020. 420–428. https://doi.org/10.1145/3336191.3371804

[5] Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. 2020. Relational Retrieval Entity Alignment. In KDD ’20: The 26th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19–23, 2020. 1095–1104. https://doi.org/10.1145/3394431.3412001

[6] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. IEEE Computer Society. 815–823. https://doi.org/10.1109/CVPR.2015.7298682

[7] Yifan Sun, Changmiao Cheng, Yuhuan Zhang, Chai Zhang, Liang Zhong, Zhaodao Wang, and Yichen Wei. 2020. Circle Loss: A Unified Perspective of Pair Similarity Optimization. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. 6597–6606. https://doi.org/10.1109/CVPR42600.2020.00262

[8] Yifan Sun, Zhefang Mao, Yuhuan Zhang, Chai Zhang, Liang Zhong, Zhaodao Wang, and Yichen Wei. 2020. Circle Loss: A Unified Perspective of Pair Similarity Optimization. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. 6597–6606. https://doi.org/10.1109/CVPR42600.2020.00262

[9] Yifan Sun, Changmiao Cheng, Yuhuan Zhang, Chai Zhang, Liang Zhong, Zhaodao Wang, and Yichen Wei. 2020. Circle Loss: A Unified Perspective of Pair Similarity Optimization. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. 6597–6606. https://doi.org/10.1109/CVPR42600.2020.00262

[10] Yifan Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding. In The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 22-27, 2017, Proceedings, Part 1 (Lecture Notes in Computer Science, Vol. 10587) Claudia d’Amato, Miriam Fernández, Mario Rodriguez, Andrea Taniguchi, Philippe Cudré-Mauroux, Juan F. Sequeda, Christos Lange, and Jeff Hefflin (Eds.). Springer, 628–644. https://doi.org/10.1007/978-3-319-68284-8_37

[11] Yifan Sun, Wei Hu, Qingsheng Zhang, and Yizhi Wang. 2019. TransEdge: Translating Relation-contextualized Embeddings for Joint Knowledge Graph. In Proceedings of the 4th International Conference on Machine Learning and Data Mining, Las Vegas, Nevada, USA, August 24–27, 2019. https://doi.org/10.18653/v1/D19-1274

[12] Yifan Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding. In The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 22-27, 2017, Proceedings, Part 1 (Lecture Notes in Computer Science, Vol. 10587) Claudia d’Amato, Miriam Fernández, Mario Rodriguez, Andrea Taniguchi, Philippe Cudré-Mauroux, Juan F. Sequeda, Christos Lange, and Jeff Hefflin (Eds.). Springer, 628–644. https://doi.org/10.1007/978-3-319-68284-8_37

[13] Yifan Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding. In The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 22-27, 2017, Proceedings, Part 1 (Lecture Notes in Computer Science, Vol. 10587) Claudia d’Amato, Miriam Fernández, Mario Rodriguez, Andrea Taniguchi, Philippe Cudré-Mauroux, Juan F. Sequeda, Christos Lange, and Jeff Hefflin (Eds.). Springer, 628–644. https://doi.org/10.1007/978-3-319-68284-8_37

[14] Yifan Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual Entity Alignment via Joint Attribute-Preserving Embedding. In The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 22-27, 2017, Proceedings, Part 1 (Lecture Notes in Computer Science, Vol. 10587) Claudia d’Amato, Miriam Fernández, Mario Rodriguez, Andrea Taniguchi, Philippe Cudré-Mauroux, Juan F. Sequeda, Christos Lange, and Jeff Hefflin (Eds.). Springer, 628–644. https://doi.org/10.1007/978-3-319-68284-8_37
[27] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. 349–357. https://doi.org/10.18653/v1/d18-1032

[28] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27–31, 2014, Quebec City, Quebec, Canada. 1112–1119. http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531

[29] Yuting Wu, Xiaolong, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. IJCAI 2019, Macao, China, August 10-16, 2019. 5278–5284. https://doi.org/10.24963/ijcai.2019/733

[30] Yuting Wu, Xiaoliu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019. 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 2019, Hong Kong, China, November 3-7, 2019. 240–249. https://doi.org/10.18653/v1/D19-1023

[31] Kun Xu, Liwei Wang, Mo Yu, Yanzong Feng, Yan Song, Zhiguo Wang, and Dong Yu. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019. Volume 1: Long Papers. 3156–3164. https://doi.org/10.18653/v1/p19-1304

[32] Hsiu-Wei Yang, Yanjun Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. 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, Hong Kong, China, November 3-7, 2019. 4430–4440. https://doi.org/10.18653/v1/D19-1451

[33] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2019. Graph Transformer Networks. In Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada. 11960–11970. http://papers.neurips.cc/paper/9367-graph-transformer-networks

[34] X. Zhao, W. Zeng, J. Tang, W. Wang, and F. Suchanek. 2020. An Experimental Study of State-of-the-Art Entity Alignment Approaches. IEEE Transactions on Knowledge and Data Engineering (2020), 1–1

[35] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2019. Neighborhood-Aware Attentional Representation for Multilingual Knowledge Graphs. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. IJCAI 2019, Macao, China, August 10-16, 2019. 1943–1949. https://doi.org/10.24963/ijcai.2019/269