Unifying Dual-Space Embedding for Entity Alignment via Contrastive Learning

Cunda Wang[1], Weihua Wang[1,2,3,*], Qiuyu Liang[1], Feilong Bao[1,2,3], Guanglai Gao[1,2,3]

[1]College of Computer Science, Inner Mongolia University, Hohhot, China
[2]National and Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Hohhot, China
[3]Inner Mongolia Key Laboratory of Multilingual Artificial Intelligence Technology, Hohhot, China

Abstract

Entity alignment (EA) aims to match identical entities across different knowledge graphs (KGs). Graph neural network-based entity alignment methods have achieved promising results in Euclidean space. However, KGs often contain complex local and hierarchical structures, which are hard to represent in a single space. In this paper, we propose a novel method named as UniEA, which unifies dual-space embedding to preserve the intrinsic structure of KGs. Specifically, we simultaneously learn graph structure embeddings in both Euclidean and hyperbolic spaces to maximize the consistency between embeddings in the two spaces. Moreover, we employ contrastive learning to mitigate the misalignment issues caused by similar entities, where embeddings of similar neighboring entities become too close. Extensive experiments on benchmark datasets demonstrate that our method achieves state-of-the-art performance in structure-based EA methods. Our code is available at `https://github.com/wonderCS1213/UniEA`.

1 Introduction

Knowledge graphs (KGs) represent real-world knowledge in the form of graphs. They typically store data in the form of triples (h, r, t), where h represents the head entity, r the relation, and t the tail entity. The completeness of KGs affects tasks such as knowledge-driven question answering (Sun et al., 2024) and recommendation (Cai et al., 2023; Liang et al., 2025). Hence, it is essential to integrate multiple source KGs to build a comprehensive KG. Entity alignment (EA) serves as an important step in this process. It aims to identify the same real-world entities referenced across different KGs.

Recently, GNN-based EA methods have achieved significant progress(Xie et al., 2023; Wang et al., 2024a; Sun et al., 2020b). However, these methods encounter two main challenges: (1) limited performance in handling complex hierarchical structures, and (2) overly similar embeddings for neighboring entities.

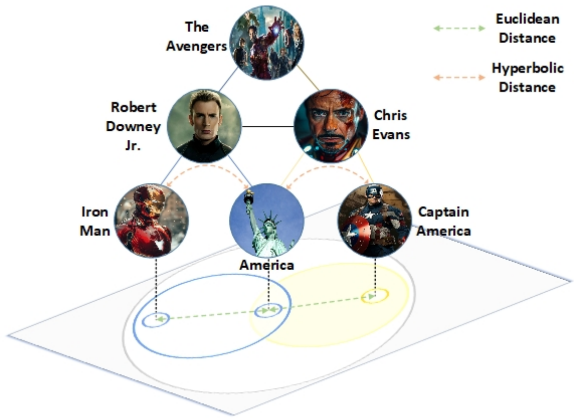


Figure 1: Knowledge graph with hierarchical structures.

As shown in Figure 1, this is a common type of hierarchical structure found in KGs. Traditional GNN-based EA methods often embed entities like “Iron Man” and “America” directly according to their Euclidean distance. Nevertheless, this fails to reflect the true distance between these two entities, leading to distortion in the graph structure embeddings. The hyperbolic space can capture the hierarchical structure of graphs (Wang et al., 2024b; Liang et al., 2024b). The hyperbolic distance better represents the true distance between the entities “Iron Man” and “America”. In addition, some methods (Wang et al., 2018; Yu et al., 2021) cause similar entities within the same KG to have embeddings that are too close in distance. For example, entities like “Robert Downey Jr.” and “Chris Evans” share multiple neighboring entities, such as “The Avengers” and “America”. Shared neighbors entities often lead to over-smoothing, which results in incorrect entity alignment. Current methods have proposed various solutions to these two challenges

*Corresponding Author. Email: wangwh@imu.edu.cn.

Proceedings of the 31st International Conference on Computational Linguistics, pages 3110–3122
January 19–24, 2025.

(Sun et al., 2020a; Guo et al., 2021; Xie et al., 2023; Wang et al., 2024a). For instance, Sun et al. (2020a) and Guo et al. (2021) explore EA task in hyperbolic space embedding and demonstrate that hyperbolic space is more effective for learning the hierarchical structure of graphs, which aids in entity alignment. Xie et al. (2023) alleviates over-smoothing through graph augmentation techniques. However, the augmentation strategies, which randomly perturb the graph topology, may degrade the quality of the graph embeddings (Shen et al., 2023). Our motivation is to consider hyperbolic space embedding as an augmentation of graph embedding. This approach not only avoids the drawbacks of traditional graph augmentation techniques but also leverages the hierarchical structure information provided by hyperbolic embedding.

To address the aforementioned issues, we propose a novel method named UniEA, which **Uni**fies the Euclidean and hyperbolic spaces embedding for **EA**. Our method is not limited to embedding in a single space. Specifically, we introduce graph attention networks (GAT) (Velickovic et al., 2018) to aggregate neighboring entities in Euclidean space, and employ hyperbolic graph convolutional networks (HGCN) (Chami et al., 2019) to learn the hierarchical structural information of the graph in hyperbolic space. We maximize the consistency between the embedding in Euclidean space and hyperbolic space through **contrastive learning**, which leads to more accurate entity embeddings. Moreover, the close distances of similar neighboring embedding severely affect the final alignment of entities. We employ **contrastive learning once again** to address the issue. The contributions of this work can be summarized as follows:

- We propose a novel EA method called UniEA. To our best knowledge, this is the first method for EA that unifying Euclidean and hyperbolic space embeddings with contrastive learning.
- We also employ contrastive learning to mitigate misalignment issues caused by overly close distances between similar entity embeddings.
- The extensive experiments on four public datasets demonstrate that our method consistently outperforms the state-of-the-art methods for structure-based EA.

2 Related work

In line with our work, we review related work in three areas: EA in Euclidean space, representation learning in hyperbolic space and improving EA with graph augmentation.

2.1 EA in Euclidean space

Current embedding-based EA methods can be broadly categorized into three types: TransE-based methods, GNN-based methods and other methods. All of these primarily aim to learn embeddings for entities and relations from relational triples.

Due to the strong performance of TransE (Bordes et al., 2013) in capturing local semantic information of entities, several methods have proposed variants of TransE for application in EA. For instance, Chen et al. (2017) addresses the inconsistency in cross-lingual embedding spaces. Zhu et al. (2017) emphasizes path information. Sun et al. (2018) treats EA as a classification task. Pei et al. (2019) enhances knowledge graph embedding by leveraging nodes with varying degrees.

TransE-based EA methods lack the ability to effectively model global structural information. As a result, recent research increasingly favors GNN-based approaches for EA. Stacking multiple GNN layers allows the model to capture information from distant neighbors, thereby facilitating the learning of global structural information. For example, Wang et al. (2018) directly stacks multiple layers of vanilla GCN (Kipf and Welling, 2017) to obtain entity embeddings. Due to the heterogeneity of KGs, the alignment performance is limited. Sun et al. (2020b) employs a gating mechanism to attempt capturing effective information from distant neighbors. MRAEA (Mao et al., 2020), RAEA (Zhu et al., 2021), KE-GCN (Yu et al., 2021), RSN4EA (Guo et al., 2019), GAEA (Xie et al., 2023), RHGN (Liu et al., 2023) , and GSEA (Wang et al., 2024a) utilize rich relational information to obtain entity embeddings. Xin et al. (2022) encoded neighbor nodes, triples and relation paths together with transformers. Unfortunately, the ability to handle complex topological structures in graphs is limited in Euclidean space.

Additionally, some methods integrate the rich information within KGs to enhance the performance of EA tasks. This includes leveraging attributes (Liu et al., 2020), entity names (Tang et al., 2020) and more (Chen et al., 2023). Jiang et al. (2024) explores the potential of large language models for EA task. Since our method focuses on structural information, we did not compare it with the aforementioned approaches to ensure fairness, consistent with prior work.

2.2 Representation learning in hyperbolic space

Hyperbolic space has recently garnered considerable attention due to its strong potential for learning hierarchical structures and scale-free characteristics. For example, Chami et al. (2019) first introduced the use of graph convolutional networks (GCN) and hyperbolic geometry through an inductive hyperbolic GCN.

Hyperbolic space representation learning has proven effective in downstream tasks such as node classification (Liang et al., 2024b) and KG completion (Liang et al., 2024a,c). Notably, existing work has successfully completed EA using hyperbolic space embedding. For example, Sun et al. (2020a) extends translational and GNN-based techniques to hyperbolic space, and captures associations by a hyperbolic transformation. Guo et al. (2021) integrates multi-modal information in the hyperbolic space and predict the alignment results based on the hyperbolic distance. Although these methods demonstrate the advantages of hyperbolic embedding, they are limited to embedding solely in hyperbolic space.

2.3 Improving EA with graph augmentation

Graph augmentation techniques primarily generate augmented graphs by perturbing the original graph through node dropout or edge disturbance, effectively enhancing the model's robustness to graph data.

Graph augmentation techniques have been proven effective in entity alignment tasks. GAEA (Xie et al., 2023) opts to generate augmented graphs by removing edges rather than adding new ones, as introducing additional edges can lead to extra noise. GSEA (Wang et al., 2024a) employs singular value decomposition to generate augmented graphs, capturing the global structural information of the graph. It leverages contrastive loss to learn the mutual information between the global and local structures of entities. However, these methods fall short in effectively learning the hierarchical structure of graphs.

3 Preliminaries

In this section, we define the EA task and explain the fundamental principles of hyperbolic space. This foundation is essential for comprehending our approach.

3.1 Entity alignment

Formally, we repesent a KG as $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ denotes entities, $\mathcal{R}$ denotes relations, $\mathcal{T} = \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ repesents triples. Given two KGs, $\mathcal{G}_1 = \{\mathcal{E}_1, \mathcal{R}_1, \mathcal{T}_1\}$ repesent source KG, $\mathcal{G}_2 = \{\mathcal{E}_2, \mathcal{R}_2, \mathcal{T}_2\}$ repesent target KG. EA aims to discern each entity pair(e_i^1, e_i^2), $e_i^1 \in \mathcal{E}_1$, $e_i^2 \in \mathcal{E}_2$ where e_i^1 and e_i^2 correspond to an identical real-world entity e_i. Typically, we use pre-aligned seed entities $\mathcal{S}$ to unify the embedding spaces of two KGs in order to predict the unaligned entities.

3.2 Hyperbolic space

Hyperbolic geometry is a non-Euclidean geometry with a constant negative curvature, where curvature measures how a geometric object deviates from a flat plane (Chami et al., 2020). In this paper, we use the d-dimensional Poincaré ball model with negative curvature $-c(c > 0)$: $H^{(d,c)} = \{x \in R^d : \parallel x \parallel^2 < \frac{1}{c}\}$. For each point $x \in H^{(d,c)}$, the tangent space (a sub-space of the Euclidean space) $T_x H_c$ is a d-dimensional vector space at point x, which contains all possible directions of path in $H^{(d,c)}$ leaving from x.

Then, we introduce exponential and logarithmic mappings in the hyperbolic space. Let α be the feature vector in the tangent space $T_o H_c$, where o is a point in the hyperbolic space $H^{(d,c)}$. Assuming o is the origin, $o = 0$, the tangent space $T_o H_c$ can be mapped to $H^{(d,c)}$ via the exponential map:

$$\exp_o^c(\alpha) = tanh(\sqrt{c}\|\alpha\|)\frac{\alpha}{\sqrt{c}\|\alpha\|}. \quad (1)$$

Conversely, the logarithmic map which maps β to $T_o H_c$ is defined as:

$$\log_o^c(\beta) = arctanh(\sqrt{c}\|\beta\|)\frac{\beta}{\sqrt{c}\|\beta\|}. \quad (2)$$

Here, β is hyperbolic space embedding.

4 Method

In this section, we elaborate on our approach in four parts. As shown in Figure 2, our method includes: 1) Euclidean space embedding, 2) hyperbolic space embedding, 3) relation encoding and fusion, and 4) the loss function.

We randomly initialize the entity and relation embedding of $\mathcal{G}_1$, represented as $\mathbf{z}_1^{\mathbb{E}} \in \mathbb{R}^{|\mathcal{E}_1| \times d_e}$ and $\mathbf{r}_1 \in \mathbb{R}^{|\mathcal{R}_1| \times d_r}$, respectively. Similarly, the entity and relation embedding of $\mathcal{G}_2$ are represented as $\mathbf{z}_2^{\mathbb{E}} \in \mathbb{R}^{|\mathcal{E}_2| \times d_e}$ and $\mathbf{r}_2 \in \mathbb{R}^{|\mathcal{R}_2| \times d_e}$. Here, $\mathbf{z}^{\mathbb{E}}$ denotes Euclidean space embedding; d_e and d_r stand for the dimensionality of entity and relation, respectively.

4.1 Euclidean space embedding

The ability of GAT to aggregate neighbor information in heterogeneous graphs has been well demonstrated (Chen et al., 2023; Wang et al., 2024a). We stack multiple layers of GAT to obtain Euclidean space embedding:

$$\begin{aligned} \mathbf{Z}^{\mathbb{E}} &= [\mathbf{z}^{(1)}, ..., \mathbf{z}^{(L)}] \\ &= GAT(\mathbf{W}_m, \mathbf{M}, \mathbf{z}^{\mathbb{E},0}), \end{aligned} \tag{3}$$

where $\mathbf{M}$ denotes the adjacency matrix, $\mathbf{W}_m \in \mathbb{R}^{d \times d}$ is a diagonal weight matrix for linear transformation.

Due to the varying importance of the neighborhoods aggregated by different layers of GAT. For example, in Figure 1, aggregating the first-order neighbors of "Chris Evans" is most beneficial. While aggregating higher-order neighbors can capture some implicit relationships of the entity, it often introduces noise. Therefore, Xie et al. (2023) introduce an attention mechanism to assign different weights to the embeddings obtained from different layers:

$$\begin{aligned} &[\hat{\mathbf{z}}^{(1)}, ..., \hat{\mathbf{z}}^{(L)}] \\ &= \mathrm{softmax}(\frac{(\mathbf{Z}^{\mathbb{E}}\mathbf{W}_q)(\mathbf{Z}^{\mathbb{E}}\mathbf{W}_k)^\top}{\sqrt{d_e}})\mathbf{Z}^{\mathbb{E}}, \end{aligned} \tag{4}$$

where $1/\sqrt{d_e}$ is the scaling factor, $\mathbf{W}_q$ and $\mathbf{W}_k$ are the learnable paramenter matrices. Finally, the Euclidean space embedding $\overline{\mathbf{z}^{\mathbb{E}}} = \frac{1}{L}\sum_{l=1}^{L} \hat{\mathbf{z}}^{(l)}$.

4.2 Hyperbolic space embedding

Our method equips HGCN (Chami et al., 2019) to learn the hierarchical structure of graphs in hyperbolic space.

Specifically, we project Euclidean space embeddings $\mathbf{z}^{\mathbb{E}}$ to hyperbolic space using exponential map (Equation 1):

$$\mathbf{z}^{\mathbb{H}} = \exp_o^c(\mathbf{z}^{\mathbb{E}}), \tag{5}$$

where $\mathbf{z}^{\mathbb{H}} \in H^{(d,c)}$, in other words, we obtain the first layer of embedding $\mathbf{z}^{\mathbb{H},0}$ in the hyperbolic space.

For the hyperbolic space embedding of the l-th layer, we can get the hyperbolic embedding of the next layer by hyperbolic feature aggregation. The hyperbolic aggregation process is as follows:

$$\mathbf{z}^{\mathbb{H},l+1} = \exp_o^c(\sigma(\mathbf{A}\log_o^c(\mathbf{z}^{\mathbb{H},l})\mathbf{W}_l)). \tag{6}$$

$\mathbf{A}$ represents the symmetric normalized adjacency matrix, σ is $ReLU(\cdot)$ and $\mathbf{W}_l$ is a trainable weight matrix.

For example, for the input $\mathbf{z}^{\mathbb{H},0}$ in 0-th layer, we can get $\mathbf{z}^{\mathbb{H},1}$ using Equation 6.

Finally, we can obtain the final output $\mathbf{z}^{\mathbb{H},L}$ in hyperbolic space. The L is a hyper-parameter denoting the number of layers of the HGCN.

4.3 Relation encoding and fusion

The same entities often share similar relations, and relational semantic information is also highly beneficial for EA. Mao et al. (2020) reveals that relying solely on the in-degree directions to accumulate neighboring information through directed edges is insufficient. Incorporating information from the out-degree directions as well would be highly beneficial. This idea facilitates the bridging and propagation of more information in graph. Hence, following this work, we use both in-degree and out-degree relation encoders to learn the representation of relations:

$$\overline{r}_{e_i} = \frac{\mathbf{A}_{e_i}^{rel_{in}}\mathbf{r}}{|N_{e_i}^{in}|} \oplus \frac{\mathbf{A}_{e_i}^{rel_{out}}\mathbf{r}}{|N_{e_i}^{out}|}, \tag{7}$$

where $|N_{e_i}^{in}|$ and $|N_{e_i}^{out}|$ are the in-degree and out-degree of e_i, respectively. $\mathbf{A}^{rel_{in}}$ denotes the adjacency matrix for in-degrees, $\mathbf{r}$ represents relation embedding.

It is worth noting that before fusion, the hyperbolic space embedding are projected to Euclidean space $\overline{\mathbf{z}^{\mathbb{H}}} = \log_o^L(\mathbf{z}^{\mathbb{H},L})$. Through the steps above, we concatenate the entity-level and relation-level features in Euclidean space to obtain the final output.

$$\tilde{\mathbf{z}^{\mathbb{H}}} = \overline{\mathbf{z}^{\mathbb{H}}} \oplus \overline{r}, \tilde{\mathbf{z}^{\mathbb{E}}} = \overline{\mathbf{z}^{\mathbb{E}}} \oplus \overline{r}. \tag{8}$$

Here, $\tilde{\mathbf{z}^{\mathbb{H}}}$ and $\tilde{\mathbf{z}^{\mathbb{E}}}$ denote final embedding in hyperbolic space and Euclidean space, respectively.

4.4 Loss function

Our loss function consists of three components: (i) a contrastive loss for aligning Euclidean and hyperbolic space embeddings $\mathcal{L}_{inter}$, (ii) an intra-graph

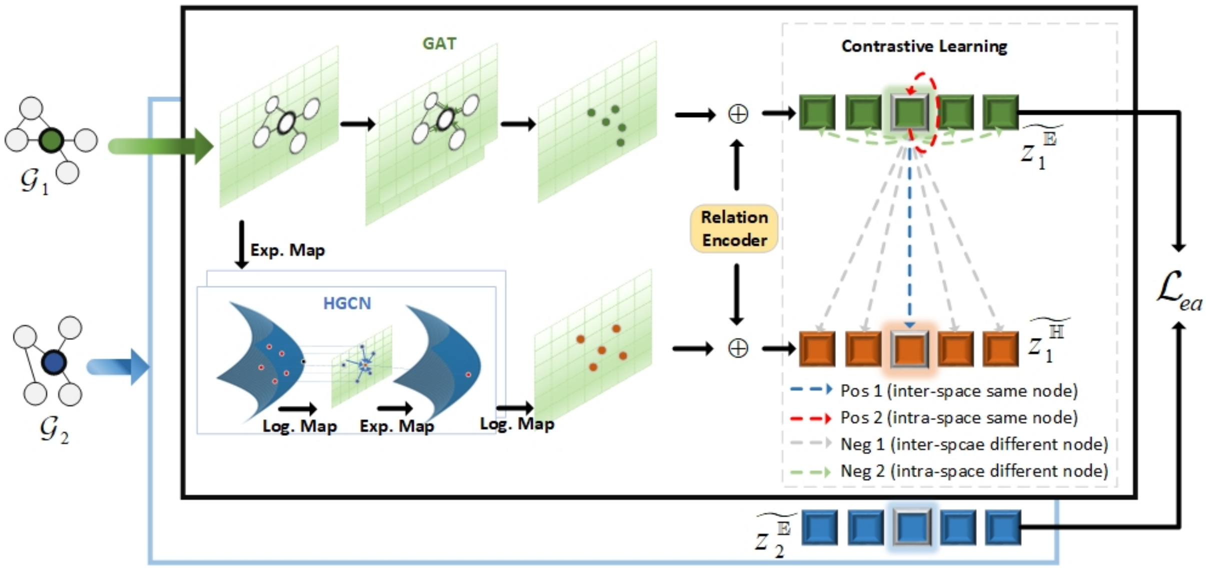


Figure 2: The framework of our proposed UniEA. Here, $\oplus$ denotes concatenate. The 'Exp. Map' operation is derived from Equation 1; the 'Log. Map' operation is derived from Equation 2.

contrastive loss to mitigate the issue of neighboring entity embeddings being too similar $\mathcal{L}_{intra}$, and (iii) a margin-based alignment loss for the entity alignment task $\mathcal{L}_{ea}$.

4.4.1 Contrastive learning loss

To ensure that the Euclidean space embedding retain their structure without distortion, we first use contrastive learning $\mathcal{L}_{inter}$ to maximize the consistency (Xie et al., 2023; Shen et al., 2023) between the Euclidean and hyperbolic space embeddings. Moreover, previous methods suggest that similar entity embedding should be closer (You et al., 2020), but being too close can negatively affect the results of EA. Therefore, we employ contrastive learning $\mathcal{L}_{intra}$, aiming to push the distances between all entities within a graph further apart. We define the contrastive learning loss as follows:

$$\mathcal{L}_{c,i}^{(G^{\mathbb{E}},G^{\mathbb{H}})} = -\log \frac{\exp(\langle \tilde{\mathbf{z}}_i^{\mathbb{E}}, \tilde{\mathbf{z}}_i^{\mathbb{H}} \rangle)}{\sum_{k\in\mathcal{E}} \exp(\langle \tilde{\mathbf{z}}_i^{\mathbb{E}}, \tilde{\mathbf{z}}_k^{\mathbb{H}} \rangle)}, \quad (9)$$

$$\mathcal{L}_{inter} = \sum_{n=\{1,2\}} \frac{1}{2|\mathcal{E}_n|} \sum_{i\in\mathcal{E}_n} (\mathcal{L}_{c,i}^{(\tilde{G}_n^{\mathbb{E}},\tilde{G}_n^{\mathbb{H}})} + \mathcal{L}_{c,i}^{(\tilde{G}_n^{\mathbb{H}},\tilde{G}_n^{\mathbb{E}})}), \quad (10)$$

$$\mathcal{L}_{intra} = \sum_{i\in\mathcal{E}} \mathcal{L}_{c,i}^{(\tilde{G}_1^{\mathbb{E}},\tilde{G}_1^{\mathbb{E}})} \quad (11)$$

Here, $\mathcal{L}_{intra}$ and $\mathcal{L}_{inter}$ can be calculated using the Equation 9.

4.4.2 Margin-based alignment loss

We use the pre-aligned entity pairs $\mathcal{S}$ to bring the embeddings of the same entities in $\mathcal{G}_1$ and $\mathcal{G}_2$ closer, while pushing the embeddings of different entities further apart. We choose to use Euclidean space embedding for the margin-based alignment loss:

$$\mathcal{L}_{ea} = \sum_{(e_i,e_j)\in S} \sum_{(e_a,e_b)\in \bar{S}_{(e_i,e_j)}} [||\tilde{\mathbf{z}}_{e_i}^{\mathbb{E}} - \tilde{\mathbf{z}}_{e_j}^{\mathbb{E}}||_{L2} + \gamma - ||\tilde{\mathbf{z}}_{e_a}^{\mathbb{E}} - \tilde{\mathbf{z}}_{e_b}^{\mathbb{E}}||_{L2}]_+, \quad (12)$$

where γ is a hyper-parameter of margin, $[x]_+ = max\{0, x\}$ is to ensure non-negative output. $\bar{S}_{(e_i,e_j)}$ is a collection of negative samples composed of randomly replaced entities e_i and e_j from the seed set $\mathcal{S}$.

4.4.3 Model training

We combine three loss functions to achieve the final training objective:

$$\mathcal{L} = \mathcal{L}_{ea} + \lambda(\mathcal{L}_{inter} + \mathcal{L}_{intra}), \quad (13)$$

where λ is a hyper-parameter to adjust the three loss functions.

5 Experiments

In this section, we conduct extensive experiments on four public datasets to demonstrate the superiority of our method. Additionally, visualization of the entity embedding from the two KGs intuitively shows that our method is more beneficial for the

EA task. Finally, we analyze the training efficiency of the method.

5.1 Experiment settings

5.1.1 Datasets

To fully demonstrate the superiority of our method, we select the OpenEA (15K-V1) dataset (Sun et al., 2020c). The dataset includes two monolingual datasets: DBpedia-to-Wikidata (D-W-15K) and DBpedia-to-YAGO (D-Y-15K), as well as two cross-lingual datasets: English-to-French (EN-FR-15K) and English-to-German (EN-DE-15K). The details of these four datasets are provided in Appendix A. The triples in OpenEA dataset consist of URLs, which not only align with the degree distribution of real-world KGs but also facilitate research on structure-based EA methods. We adhere to the data split ratios used in prior studies (Sun et al., 2020c), where 20% of the alignments are used for training, 10% for validation, and 70% for testing. We report the average results of five-fold cross-validation. The results for each fold are presented in Appendix B.

5.1.2 Implement details

Our experiment conducted with a single NVIDIA 4090 GPU with 24GB of memory. We initialize the trainable parameters with Xavier initialization (Glorot and Bengio, 2010) and optimize the loss using Adam (Kingma and Ba, 2015). Regarding hyperparameters, the entity dimension is set to 256, and the relation dimension is set to 32. The margin for the alignment loss is set to 1. We perform a grid search to determine the optimal parameter λ for the final training objective, selecting from the set {0.1, 1, 10, 100, 300, 1000}. The details of λ are provided in Appendix C. GAT and HGCN both utilize a two-layer network. We generate 5 negative samples for each positive sample. During inference, we use Cross-domain Similarity Local Scaling (Lample et al., 2018) to post-process the cosine similarity matrix, which is employed by default in some recent works (Sun et al., 2020c; Liu et al., 2023).

We use H@k and MRR as evaluation metrics to assess our method, with higher values indicating a greater number of correctly matched entities. We select k values of $\{1, 5\}$.

5.1.3 Baseline

To comprehensively evaluate the superiority of our method, we categorize our baselines into three groups: TransE-based, GNN-based, and related methods.

- TransE-based methods. These methods leverage variants of TransE to model each triple individually, utilizing strong local structural information: MTransE (Chen et al., 2017), IPTransE (Zhu et al., 2017), AlignE (Sun et al., 2018), and SEA (Pei et al., 2019).
- GNNs-based methods. These methods aggregate neighborhood information by stacking multiple layers of networks: GCN-Align (Wang et al., 2018), AliNet (Sun et al., 2020b), KE-GCN (Yu et al., 2021), and RHGN (Liu et al., 2023).
- Related methods. We classify these four methods into one category: HyperKA (Sun et al., 2020a) is the method for EA in hyperbolic space. GAEA (Xie et al., 2023) uses edge deletion information for graph augmentation. IMEA (Xin et al., 2022) is a strong baseline that combines information from nodes, triples, and relation paths. GSEA (Wang et al., 2024a) uses singular value decomposition of the adjacency matrix to obtain the global structural information of the entities.

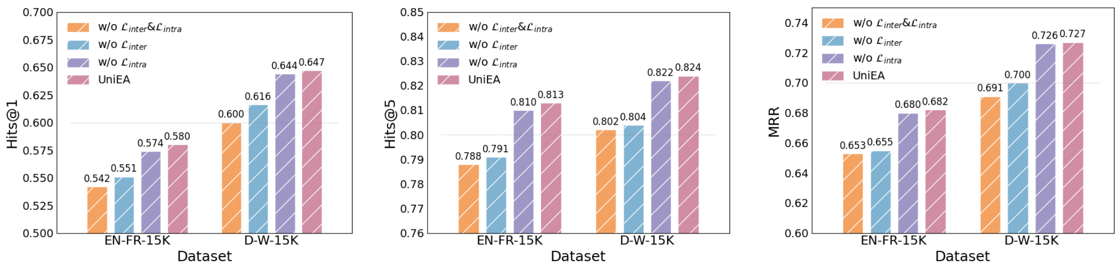
5.2 Main results

The results of all methods on OpenEA datasets are shown in Table 1. Our method outperforms all other methods. We conducted an analytical comparison with baseline methods. Since TransE and SEA cannot effectively learn the structural features of the graph, even with semi-supervised strategies, they fail to improve alignment accuracy. HyperKA operates in hyperbolic space, but since the hierarchical structure of the four datasets is not particularly pronounced, its performance is inferior to some Euclidean space methods.

Methods that utilize relational semantic information, such as GSEA, IMEA , GAEA, and RHGN, significantly outperform others and indicate that relationships are highly beneficial for entity alignment. GAEA uses contrastive learning on different views, but its performance is unstable. Our method is not limited by the drawbacks of traditional graph augmentation methods. It not only captures neighborhood information in Euclidean space but also learns the hierarchical structure in hyperbolic space. Our experiments demonstrate that learning the hierarchical structure in knowledge graphs is crucial.

Methods	EN-FR-15K			EN-DE-15K			D-W-15K			D-Y-15K		
	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR
MtransE	.247	.467	.351	.307	.518	.407	.259	.461	.354	.463	.675	.559
IPTransE	.169	.320	.243	.350	.515	.430	.232	.380	.303	.313	.456	.378
AlignE	.357	.611	.473	.552	.741	.638	.406	.627	.506	.551	.743	.636
SEA	.280	.530	.397	.530	.718	.617	.360	.572	.458	.500	.706	.591
GCN-Align	.338	.589	.451	.481	.679	.571	.364	.580	.461	.465	.626	.536
AliNet	.364	.597	.467	.604	.759	.673	.440	.628	.522	.559	.690	.617
KE-GCN	.408	.670	.524	.658	.822	.730	.519	.727	.608	.560	.750	.644
RHGN*	.500	.739	.603	.704	.859	.771	.560	.753	.644	.708	.831	.762
HyperKA	.353	.630	.477	.560	.780	.656	.440	.686	.548	.568	.777	.659
IMEA	.458	.720	.574	.639	.827	.724	.527	.753	.626	.639	.804	.712
GAEA*	.548	.783	.652	.731	.887	.800	.618	.802	.802	.671	.802	.731
GSEA*	.561	.803	.669	.740	.893	.807	.628	.819	.713	.694	.836	.758
UniEA(ours)	**.580**	**.811**	**.682**	**.748**	**.898**	**.813**	**.648**	**.826**	**.728**	**.712**	**.841**	**.771**

Table 1: Entity alignment result of OpenEA Datasets. The best result in each column is highlighted in bold. * indicates results reproduced from their source code, while other experimental results are from Xie et al. (2023). For fairness in the experimental results, we modified the GAEA code to use CSLS during the inference phase.



(a) H@1 (b) H@5 (c) MRR

Figure 3: The results of ablation experiment on EN-FR-15K and D-W-15K.

5.3 Ablation studies

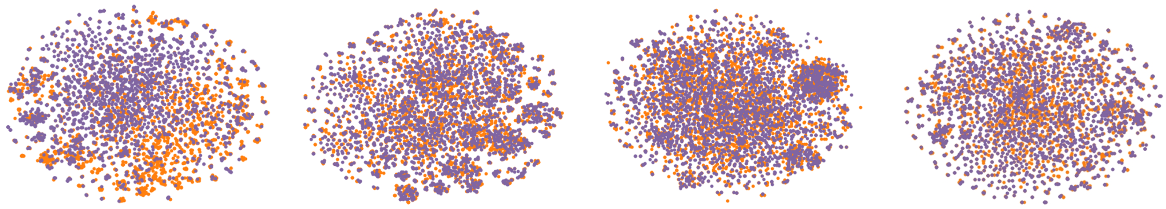
We conducted ablation experiments on EN-FR-15K and D-W-15K by removing $\mathcal{L}_{inter}$, $\mathcal{L}_{intra}$, and $\mathcal{L}_{inter}\&\mathcal{L}_{intra}$. As shown in Figure 3, all metrics decreased after the removal of each module, demonstrating the effectiveness of each component.

- **w/o** $\mathcal{L}_{inter}$: By removing the learning in hyperbolic space and using GAT to aggregate neighborhood information and learn relational semantics in Euclidean space, we observed a decline in all metrics across the datasets. This indicates that learning both Euclidean and hyperbolic embedding through contrastive learning is effective.
- **w/o** $\mathcal{L}_{intra}$: Removing contrastive learning within the $\mathcal{G}_1$ in Euclidean space also resulted in a decline in the performance. Our method shows the most significant decline and effectively pushes similar but easily confused entities further apart in the embedding space.
- **w/o** $\mathcal{L}_{inter}\&\mathcal{L}_{intra}$: The decline observed after removing it demonstrates that combining $\mathcal{L}_{intra}$ and $\mathcal{L}_{inter}$ is beneficial for enhancing alignment performance.

5.4 Visualization of Entity Embedding

To more intuitively highlight the performance of our method, we use t-SNE (Rauber et al., 2016) to visualize the entity embedding. In the baseline, we categorized all comparison methods into three groups, and then, we selected the top three methods for comparison. We randomly selected 3,000 entity pairs, and the final embeddings are shown in Figure 4.

The embedding from AlignE (Figure 4(a)) are



(a) AlignE (b) GAEA (c) RHGN (d) UniEA(ours)

Figure 4: Visualization of entity embedding on EN-FR-15K. Different colors represent different KGs.

clearly the worst, with one KG's embeddings concentrated in the upper left and the other in the lower right. As a result, during the alignment inference phase, it fails to match the correct entities. GAEA (Figure 4(b)) shows multiple clusters along the edges, with the left half being sparser than the right. The distribution of entity embedding is uneven, which causes similar entities to be placed too close together, leading to incorrect alignments. RHGN (Figure 4(c))shows a large cluster in the upper right corner, with uneven distribution and poor embedding results. In contrast, our method (Figure 4(d)) exhibits a uniform distribution without noticeable clustering. Observing the surrounding area, the points of different colors in our method overlap completely and indicate better embedding performance. Therefore, our method achieves results superior to other methods.

Methods	#Params (M)
AliNet	∼16.18M
IMEA	∼20.44M
GAEA	∼8.10M
RHGN	∼8.62M
UniEA(ours)	∼15.86M

Table 2: Methods parameters comparison

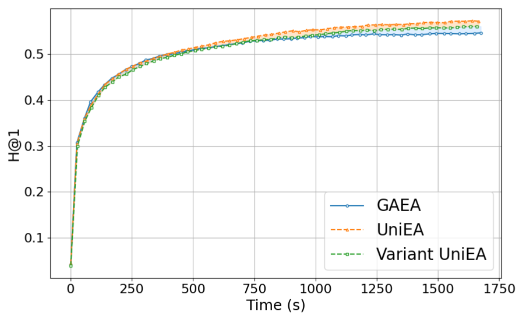


Figure 5: H@1 results and training times on EN-FR-15K.

5.5 Auxiliary Experiments

5.5.1 Parameter sizes analysis

We selected four baseline models for comparison of parameter sizes, as shown in Table 2. GAEA uses a single GAT network for training, significantly reducing its model complexity. Our method requires different networks for training in both Euclidean and hyperbolic spaces, which causes the parameter size of our model to be nearly double that of GAEA. However, our parameter size is much smaller than that of IMEA, which uses complex features. Despite this, our method outperforms all structure-based methods.

5.5.2 Efficiency analysis

To evaluate the time efficiency of our method, we conducted a comparative analysis with GAEA. We also included our variant, UniEA-w/o $\mathcal{L}_{intra}$, which does not use contrastive learning for similar entities. The final results are shown in Figure 5. For a fair comparison, we ran 300 epoch and used the same entity embedding dimension of 256 and relation embedding dimension of 32.

GAEA updates the augmented graph every 10 epochs, which impacts its training efficiency. In contrast, our method does not require augmented graph updates, making it the faster. However, since the UniEA method introduces contrastive learning to push similar entities further apart, it requires longer training time than variant UniEA.

6 Conclusion

In this paper, we propose a novel method for EA that unifies dual-space through contrastive learning. We take the learning in hyperbolic space as a specialized form of graph augmentation. This study focuses on maximizing the consistency between Euclidean and hyperbolic space embeddings through contrastive learning. Additionally, we employ contrastive learning to increase the distance between embedding of similar entities in Euclidean space, thereby preventing erroneous alignments caused by similarity. Finally, we conduct analyses through ablation studies, visualizations, parameter size comparisons, and evaluations of time efficiency.

Limitation

We acknowledge three limitations in our method. First, in the Auxiliary Experiments, we discuss parameter size and time efficiency. While our approach addresses the drawbacks of traditional contrastive learning in generating contrastive views, the introduction of a hyperbolic convolutional network results in a significant number of parameters. Second, our method focuses solely on learning structural and relational information, leaving a wealth of attribute information within KGs untapped. Finally, real-world KGs are predominantly unlabeled, and labeling data is costly. We lack unsupervised or semi-supervised strategies to enhance alignment performance.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Nos.62066033); Inner Mongolia Natural Science Foundation (Nos.2024MS06013, 2022JQ05); Inner Mongolia Autonomous Region Science and Technology Programme Project (Nos.2023YFSW0001, 2022YFDZ0059, 2021GG0158); We also thank all anonymous reviewers for their insightful comments.

References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 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*, pages 2787–2795.

Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. Lightgcl: Simple yet effective graph contrastive learning for recommendation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6901–6914. Association for Computational Linguistics.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4869–4880.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1511–1517. ijcai.org.

Zhuo Chen, Jiaoyan Chen, Wen Zhang, Lingbing Guo, Yin Fang, Yufeng Huang, Yichi Zhang, Yuxia Geng, Jeff Z. Pan, Wenting Song, and Huajun Chen. 2023. Meaformer: Multi-modal entity alignment transformer for meta modality hybrid. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 3317–3327. ACM.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org.

Hao Guo, Jiuyang Tang, Weixin Zeng, Xiang Zhao, and Li Liu. 2021. Multi-modal entity alignment in hyperbolic space. *Neurocomputing*, 461:598–607.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2505–2514. PMLR.

Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, and Yuanzhuo Wang. 2024. Unlocking the power of large language models for entity alignment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7566–7583. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Qiuyu Liang, Weihua Wang, Feilong Bao, and Guanglai Gao. 2024a. Fully hyperbolic rotation for knowledge graph embedding. In *ECAI 2024 - 27th European Conference on Artificial Intelligence, 19-24 October 2024, Santiago de Compostela, Spain - Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, volume 392 of *Frontiers in Artificial Intelligence and Applications*, pages 1615–1622. IOS Press.

Qiuyu Liang, Weihua Wang, Feilong Bao, and Guanglai Gao. 2024b. L\ ^{2}gc: Lorentzian linear graph convolutional networks for node classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 9988–9998. ELRA and ICCL.

Qiuyu Liang, Weihua Wang, Lei Lv, and Feilong Bao. 2025. Knowledge graph-enhanced recommendation with box embeddings. In *Chinese Computational Linguistics*, pages 274–288, Singapore. Springer Nature Singapore.

Qiuyu Liang, Weihua Wang, Jie Yu, and Feilong Bao. 2024c. Effective knowledge graph embedding with quaternion convolutional networks. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 183–196. Springer.

Xukai Liu, Kai Zhang, Ye Liu, Enhong Chen, Zhenya Huang, Linan Yue, and Jiaxian Yan. 2023. RHGN: relation-gated heterogeneous graph network for entity alignment in knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8683–8696. Association for Computational Linguistics.

Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, and Tat-Seng Chua. 2020. Exploring and evaluating attributes, values, and structures for entity alignment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6355–6364. Association for Computational Linguistics.

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*, pages 420–428. ACM.

Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. 2019. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 3130–3136. ACM.

Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. 2016. Visualizing time-dependent data using dynamic t-sne. In *18th Eurographics Conference on Visualization, EuroVis 2016 - Short Papers, Groningen, The Netherlands, June 6-10, 2016*, pages 73–77. Eurographics Association.

Xiao Shen, Dewang Sun, Shirui Pan, Xi Zhou, and Laurence T. Yang. 2023. Neighbor contrastive learning on learnable graph augmentation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 9782–9791. AAAI Press.

Yanhan Sun, Zhenfang Zhu, Zicheng Zuo, Kefeng Li, Shuai Gong, and Jiangtao Qi. 2024. DSAMR: dual-stream attention multi-hop reasoning for knowledge-based visual question answering. *Expert Syst. Appl.*, 245:123092.

Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020a. Knowledge association with hyperbolic knowledge graph embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5704–5716. Association for Computational Linguistics.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4396–4402. ijcai.org.

Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020b. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 222–229. AAAI Press.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai

Li. 2020c. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proc. VLDB Endow.*, 13(11):2326–2340.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2020. BERT-INT: A bert-based interaction model for knowledge graph alignment. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3174–3180. ijcai.org.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Cunda Wang, Weihua Wang, Qiuyu Liang, Jie Yu, and Guanglai Gao. 2024a. GSEA: global structure-aware graph neural networks for entity alignment. In *Natural Language Processing and Chinese Computing - 13th National CCF Conference, NLPCC 2024, Hangzhou, China, November 1-3, 2024, Proceedings, Part II*, volume 15360 of *Lecture Notes in Computer Science*, pages 187–199. Springer.

Weihua Wang, Qiuyu Liang, Feilong Bao, and Guanglai Gao. 2024b. Distance-adaptive quaternion knowledge graph embedding with bidirectional rotation. *arXiv preprint arXiv:2412.04076*.

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*, pages 349–357. Association for Computational Linguistics.

Feng Xie, Xiang Zeng, Bin Zhou, and Yusong Tan. 2023. Improving knowledge graph entity alignment with graph augmentation. In *Advances in Knowledge Discovery and Data Mining - 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25-28, 2023, Proceedings, Part II*, volume 13936 of *Lecture Notes in Computer Science*, pages 3–14. Springer.

Kexuan Xin, Zequn Sun, Wen Hua, Wei Hu, and Xiaofang Zhou. 2022. Informed multi-context entity alignment. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 1197–1205. ACM.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Donghan Yu, Yiming Yang, Ruohong Zhang, and Yuexin Wu. 2021. Knowledge embedding based graph convolutional network. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1619–1628. ACM / IW3C2.

Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4258–4264. ijcai.org.

Renbo Zhu, Meng Ma, and Ping Wang. 2021. RAGA: relation-aware graph attention networks for global entity alignment. In *Advances in Knowledge Discovery and Data Mining - 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11-14, 2021, Proceedings, Part I*, volume 12712 of *Lecture Notes in Computer Science*, pages 501–513. Springer.

Appendix

A Dataset Statistics

Table 3 provides rich information about OpenEA datasets.

B Our methods result of OpenEA Datasets

We used the same parameters for 5-fold cross-validation for each dataset. The reported result UniEA(avg.) in Table 4 is obtained by averaging over five-fold.

C Hyper-parameter settings.

We conducted a single experiment with fold 4, and ultimately chose $\lambda = 300$ for the EN-FR-15K, EN-DE-15K, and D-W-15K datasets, while selecting $\lambda = 100$ for the D-Y-15K dataset. The experimental results are shown in Table 5.

Dataset	EN-FR-15K		EN-DE-15K		D-W-15K		D-Y-15K	
	English	French	English	German	DBpedia	Wikidata	DBpedia	YAGO
#Ent.	15,000	15,000	15,000	15,000	15,000	15,000	15,000	15,000
#Rel.	267	210	215	131	248	169	165	28
#Rel tr.	47,334	40,864	47,676	50,419	38,265	42,746	30,291	26,638

Table 3: The statistics of OpenEA datasets

Fold	EN-FR-15K			EN-DE-15K			D-W-15K			D-Y-15K		
	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR
1	.580	.807	.680	.744	.901	.812	.654	.831	.733	.712	.838	.770
2	.576	.808	.679	.748	.895	.813	.652	.829	.731	.716	.845	.774
3	.585	.816	.687	.752	.902	.817	.646	.824	.727	.717	.847	.775
4	.580	.810	.682	.746	.895	.811	.647	.824	.727	.710	.838	.768
5	.582	.818	.685	.751	.899	.815	.644	.823	.723	.709	.840	.769
UniEA(avg.)	.580	.811	.682	.748	.898	.813	.648	.826	.728	.712	.841	.771

Table 4: Entity alignment result of OpenEA datasets on every fold.

λ	EN-FR-15K			EN-DE-15K			D-W-15K			D-Y-15K		
	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR	H@1	H@5	MRR
0.1	.544	.787	.652	.712	.883	.787	.603	.806	.693	.672	.824	.740
1	.544	.790	.653	.715	.879	.788	.603	.807	.694	.672	.828	.742
10	.547	.795	.657	.718	.882	.791	.616	.817	.704	.688	.834	.754
100	.572	.811	.678	.741	.895	.809	.641	.830	.725	.710	.838	.768
300	.580	.810	.682	.746	.895	.811	.647	.824	.727	.703	.824	.759
500	.577	.806	.679	.744	.893	.810	.641	.820	.721	.690	.819	.749
1000	.572	.793	.670	.742	.887	.806	.634	.806	.711	.682	.820	.745

Table 5: Experimental results with different hyper-parameters λ.