


Research Article

Complex Representation Learning with Graph Convolutional Networks for Knowledge Graph Alignment

Darnbi Sakong ¹, Thanh Trung Huynh ², Thanh Tam Nguyen ¹,
Thanh Toan Nguyen ³, Jun Jo ¹ and Quoc Viet Hung Nguyen ¹

¹Griffith University, Gold Coast, Australia

²École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

³Faculty of Information Technology, HUTECH University, Ho Chi Minh City, Vietnam

Correspondence should be addressed to Thanh Tam Nguyen; thanhtamlhp@gmail.com

Received 30 November 2022; Revised 7 September 2023; Accepted 15 November 2023; Published 2 December 2023

Academic Editor: Vasudevan Rajamohan

Copyright © 2023 Darnbi Sakong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The task of discovering equivalent entities in knowledge graphs (KGs), so-called KG entity alignment, has drawn much attention to overcome the incompleteness problem of KGs. The majority of existing techniques learns the pointwise representations of entities in the Euclidean space with translation assumption and graph neural network approaches. However, real vectors inherently neglect the complex relation structures and lack the expressiveness of embeddings; hence, they may guide the embeddings to be falsely generated which results in alignment performance degradation. To overcome these problems, we propose a novel KG alignment framework, ComplexGCN, which learns the embeddings of both entities and relations in complex spaces while capturing both semantic and neighborhood information simultaneously. The proposed model ensures richer expressiveness and more accurate embeddings by successfully capturing various relation structures in complex spaces with high-level computation. The model further incorporates relation label and direction information with a low degree of freedom. To compare our proposal against the state-of-the-art baseline techniques, we conducted extensive experiments on real-world datasets. The empirical results show the efficiency and effectiveness of the proposed method.

1. Introduction

The knowledge graph (KG) is directed graph-structured data that captures intricate relationships of real-world objects. KGs cover a wide range of real-world facts in the form of triples to make them machine-readable. Following the resource description format, a triple is a single unit in KG and it can be represented in the form of $\langle \text{subject}, \text{predicate}, \text{object} \rangle$ which mirrors human language structure. More specifically, it can be categorized into relation triple $\langle \text{entity}_1, \text{relation}, \text{entity}_2 \rangle$ and attribute triple $\langle \text{entity}, \text{attribute type}, \text{attribute value} \rangle$. Taking an example from DBpedia, $\langle \text{Berlin}, \text{IsCapitalOf}, \text{Germany} \rangle$ is one of the relation triples and $\langle \text{Berlin}, \text{areaCode}, 030 \rangle$ is an attribute triple. Due to the recent growth of the Linked Open Data, KGs such as DBpedia [1], ConceptNet [2], and WordNet have rapidly proliferated in a variety of domains and languages by enabling AI upstream and downstream

applications (e.g., question-answering [3], recommendation system [4], and semantic search [5]) to harness abundant machine-readable knowledge-powered data.

Despite its advancement, there still remain many challenges such as *incompleteness* due to the ever-evolving nature of KGs under the open-world assumptions. Besides, KGs are often developed for different purposes by individual experts and even in heterogeneous human languages, which widens the knowledge gap between them. As a solution for bridging the knowledge gap, KG alignment techniques aim to find the corresponding entities between multiple KGs, also known as *anchor links*, to complement each other. However, only a limited number of anchor links exist in real-world cases, and finding latent links manually is costly and can be easily misguided.

To handle such tasks automatically in an efficient way, a line of study has been conducted to develop effective KG

alignment techniques by adopting machine learning methods. The most renowned techniques are based on the *representation learning* approach which represents resources of KGs as points in multidimensional real coordinate spaces. The main goal of such techniques is to retain the expressiveness of each point, so-called *embeddings*, as much as possible by extracting salient alignment-oriented features. Given partially aligned KGs, the alignment model learns meaningful embeddings to automatically discover the links between the equivalent entities (or relations) based on their pairwise distances in the vector space. The representation learning-based KG alignment techniques can be divided into two main categories: *semantic-based* and *neighborhood structure-based* techniques. *Semantic-based* [6–8] techniques concentrate on capturing the semantic information contained in the triples. One representative method used for learning the semantic pattern of triples is TransE [9] which considers the relation in triples as translation from head entity to tail entity. In recent years, a number of state-of-the-art graph neural networks (GNNs) [10–12] have approaches such as GCN which have been largely utilized for capturing the structural information in the KGs. The fundamental intuition of GNNs is to incorporate local neighborhood information for representing entity embeddings. They have demonstrated their robustness in learning neighborhood-aware vector representations with various message-passing schemes.

However, there is still much room for improvement regarding the following challenges:

- (i) *Relation Patterns and Structure*. There are some significant challenges to be addressed for semantic-based techniques which use the translation assumption. First, the translation assumption cannot cover triangular structure (e.g., cyclic) since all the relation embeddings involved have to converge to 0 which results in a contradiction. Second, the same contradiction occurs for capturing symmetry patterns of relations which frequently appear in practice [13] and has shown its robustness at capturing triangular relation structures; however, it requires extra graphs which represent the structure of the relations, hence resulting in a scalability problem when it comes to large-scale KGs. These listed limitations can lead the embeddings to learn incorrect semantic information which leads the encoded vectors to be obsolete. Thus, to learn better representations for KGs, it is stressed to carefully model such intricate relation structures with a more efficient and scalable approach.
- (ii) *Relation-Aware Neighborhood Information*. GNN-based models have shown their superiority in learning local neighborhood structure; however, they often overlook to consider several important factors such as the multiple perspectives of relational information. For example, the adjacency matrix that propagated through GCN layers only preserves simple connection information while losing informative indicators by ignoring other

additional relational information (e.g., direction, pattern, and relation type). Existing works [14, 15] attempt to incorporate type and direction information of relations; however, such techniques either rely on abstract concepts or suffer from a high degree of freedom. Moreover, it exclusively learns representations of entities while embeddings of relations play a significant role in many tasks.

- (iii) *Expressiveness of Embeddings*. Existing GCNs are limited to being trained in Euclidean space where the vectors are represented with real numbers. However, it has been manifested by recent studies that the embeddings in complex space have more potential to retain rich feature information compared to those in real vector space [16–18] by using complex computations. For example, unlike the dot product in real vector space, the Hermitian product in complex space can capture higher-order features of relation based on the antisymmetric nature of it [17]. Hence, the GCN that operates in the complex space is worth to be explored to go beyond the existing works.

To tackle the aforementioned issues, we propose a complex embedding-based model combined with a novel complex GCN framework, namely, ComplexGCN. ComplexGCN takes advantage from two modules: *rotation-based embedding module* (RE) and *deep integration module* (DI) to complement two respective perspectives of *knowledge semantic* and *neighborhood information* simultaneously by learning embeddings in complex space. RE learns alignment-oriented complex vector representations in complex space \mathbb{C}^n of both entities and relations with rotation constraints. By leveraging the rotation concept, it can inherently model all of the relation patterns (e.g., hierarchy, symmetry, and composition) resulting in better semantic-preserving embeddings compared to the translation assumption. We further discover that cyclic structures such as a set of three triples (e.g., $\langle e_1, r, e_2 \rangle$, $\langle e_2, r, e_3 \rangle$, $\langle e_3, r, e_1 \rangle$) can be captured properly with the rotation assumption. With the rotation score function, we fuse two loss functions to encourage the equivalent entities to be generated close to each other. We further combine DI with RE to consider both semantic and neighborhood information. Different from vanilla GCN, DI has the capability to generate both entity and relation embeddings while keeping track of multiple perspectives of relation information (e.g., direction and relation type). It is worth noting that our proposal prevents overparameterization with the weight-sharing mechanism. Also, it effectively passes the complex embeddings through GCN layers with the Hadamard product. Therefore, our model successfully preserves the expressiveness of the embeddings with more meaningful and intense computation. Note that, we not only train the embeddings based on alignment likelihood but also the plausibility of the triples via the end-to-end learning framework of the DI module. After generating two separate embedding spaces with SE and DI, the alignment result is then estimated by combining the alignment scores generated by the respective module.

1.1. Contributions. The main contributions of this paper can be summarized as follows:

- (i) ComplexGCN learns the representation of both entities and relations in complex spaces from the two viewpoints: semantic-based (RE module) and neighborhood information (DI).
- (ii) RE module generates knowledge semantic-preserving embeddings in complex space to better capture both triangular structure and relation patterns [16] than conventional translation assumption. We further prove that Hadamard products can capture not only relation patterns but also cyclic structure in KGs.
- (iii) DI module successfully incorporates additional relation information (e.g., direction and type) while propagating KGs through GCN layers. To surpass the expressiveness of real vector representations, it learns embeddings with complex numbers with an efficient calculation method inspired by QGNN [19].
- (iv) Different from existing methods, our DI module requires simpler equations and fewer parameters. Moreover, it is a pioneering approach to model the representation learning of knowledge graphs in complex spaces. Different from complex-based GNNs [19, 20], our DI module incorporates directional, self-loop, and relational information that are specific for knowledge graphs.
- (v) The fusion of RE and DI modules guarantees to retain the semantic and topological information at the same time by complementing each other. The vanishing problem of semantic features of DI is supplemented by the RE, while the lack of structural information of the RE module is filled up with DI.
- (vi) We conduct extensive experiments on popular publicly available real-world datasets to evaluate ComplexGCN with state-of-the-art baselines. The empirical results demonstrate that our model produces the best performance against the baselines.

The remainder of this paper is organized as follows. The motivations of our work are discussed in Section 2. In Section 3, the problem is formulated and a succinct overview of our framework is presented. In Section 4 and Section 5, the details of our proposal are demonstrated. Our empirical results are organized in Section 6. Finally, the related works are reviewed in Section 7, and Section 8 concludes the paper.

2. Preliminaries

2.1. Motivations. In Figure 1, we offer an example of a KG entity alignment problem containing four main relation patterns: symmetry, inversion, composition, and cyclic patterns. Furthermore, the KG in the upper container is English-version KG, and the one in the lower container is German-version KG, respectively. The black solid arrow connecting entities in different KGs depicts seed alignments

(e.g., *owl:sameAs*), a solid arrow connecting entities in a single KG is an example of typical relation, and all the dotted arrows or entities in a single KG are newly discovered resources. As we can observe from the illustration, there are four types of relation patterns depicted in different colors of arrows: a set of green arrows depicts a composition pattern, a set of pink arrows describes an inversion pattern, blue arrows represent a symmetry pattern, and a set of orange arrows shows the cyclic structure.

It is crucial to carefully encode these complex relation patterns into the relation embeddings since the former works have proven that the local semantic information greatly impacts on the meaningful alignment-oriented embeddings [6, 7, 9]. However, the score function that is constrained by translation assumption [9] has its limitations at coping with these relation patterns. For example, when we attempt to encode the semantics of three triples $\langle \text{Michael_Jackson}, \text{Is Brother Of}, \text{Jackie_Jackson} \rangle$, $\langle \text{Jackie_Jackson}, \text{Is Brother Of}, \text{Tito_Jackson} \rangle$, and $\langle \text{Tito_Jackson}, \text{Is Brother Of}, \text{Michael_Jackson} \rangle$ based on the translation assumption in a 2-dimensional Euclidean space, it faces discrepancy in relation embedding of relation *Is Brother Of* by producing multiple embeddings for it. Such a problem can cause the inconsistency of embeddings and can seriously harm the aligners by hindering to capture of semantic information.

2.2. A Solution Based on Complex Vector Space. In recent years, the majority of representation learning methods have focused on generating embeddings in Euclidean space where each element of a vector is a real number. However, the computations in Euclidean space have shown their limitations in capturing diverse facets of KGs. One remedy for this problem is that recent studies have been started to utilize complex space where each entry of a vector is composed of a complex number with the form, $a + bi$. By involving additional imaginary parts, complex vector representations enable more expressive rendering with intense computations.

In particular, instead of using a real-valued space, entities and relations are represented in a complex space, in which for example the head entity has a real part and an imaginary part [21]. ComplEx [17] introduced a complex vector space that can capture both symmetric and anti-symmetric relations using the Hermitian dot product for the composition of the relation, head, and the conjugate of the tail. Inspired by Euler's identity, RotatE [16] proposed a rotational model that considers the relation as a rotation from head entity to tail entity in the complex space, where the elementwise Hadamard product is used for composition. QuatE [22] extends the complex-valued space to a hyper-complex space by using a quaternion that has three imaginary components, and the Hamilton product is used as the compositional operator for the head entity and relation. Both RotatE and QuatE can capture inversion and composition patterns, as well as symmetry and antisymmetry. However, QuatE incurs more computational overhead than RotatE, as it uses the Hamilton product to capture latent

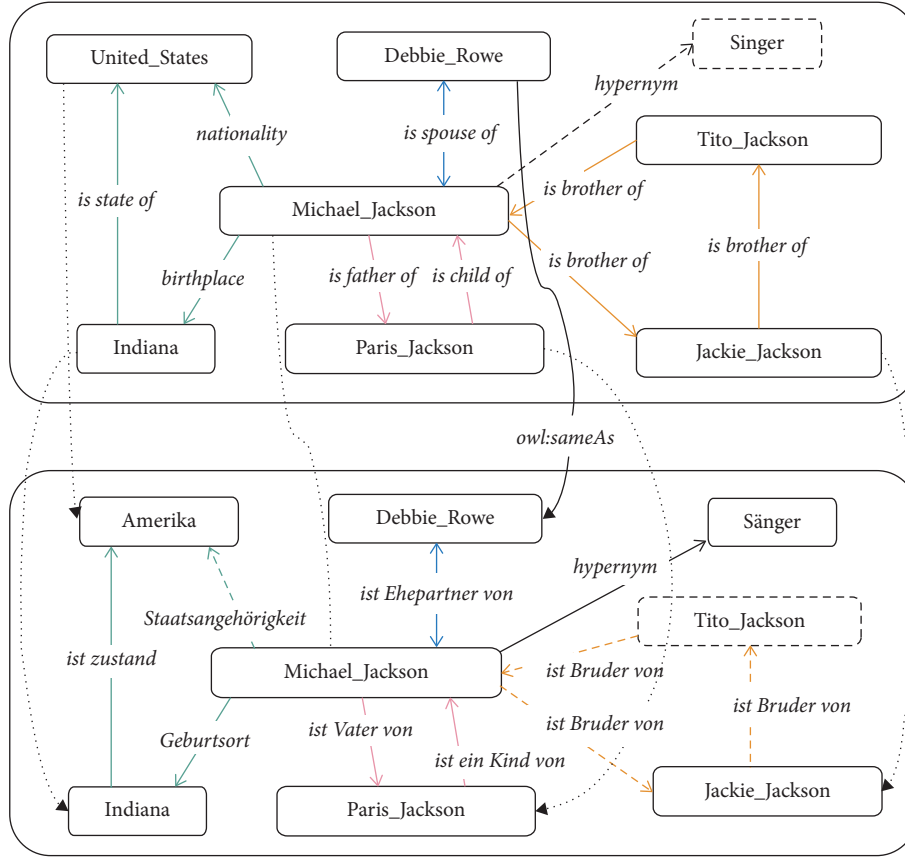


FIGURE 1: Example of entity alignment between KGs with four relationship patterns. The upper KG represents the English-version KG, and the lower one is the toy graph of the German-version KG.

interdependencies within the 4-D space of entities and relations. Especially for the KG representation learning tasks, complex vector representations successfully reflect the complicated relationships of entities compared to real embeddings. Starting from ComplEx [17], RotatE [16] to MRotatE [18], encodings of complex embedding for KG have been explored extensively by applying various calculus and measurements. We further develop our model by generating complex embeddings which capture not only the semantic information but also the topological information of KGs.

One remedy for this issue can be the representation learning method in complex spaces. Based on the expressive computation using the Hadamard product in complex spaces [16], we can successfully capture all four relation patterns based on the rotation approach resulting in semantic well-preserved embeddings. For instance, in Figure 2, the rotation assumption-based RotatE [16] in complex space can capture cyclic relation patterns which have been neglected by the translation assumption in Euclidean space. Despite the significance of the rotation-based score function, only resorting to the local semantic information can lead the model to neglect the global scale of information. To consider both local and global information, the GNN mechanism that works in complex spaces is highly needed. We believe that a reliable and promising alignment can be performed by

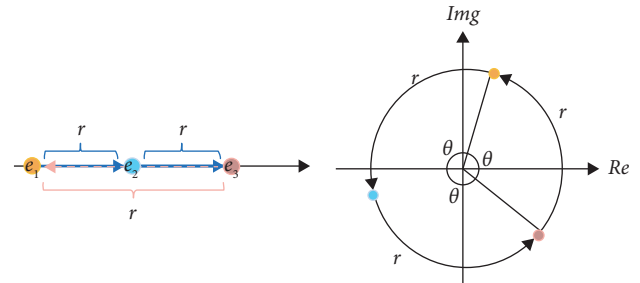


FIGURE 2: Examples of TransE and RotatE attempting to capture cyclic structure in 1-dimensional real and complex spaces. The x -axis denotes the real axis, and the y -axis denotes the imaginary axis in complex space. The picture demonstrates how RotatE captures cyclic structure, while TransE fails to capture it.

fusing the semantic-preserving and neighborhood-aware methods that operate in complex spaces.

3. Problem Statement

3.1. Knowledge Graph Structure. Before getting to the formulation, some necessary notations are provided as follows:

Resources. Resources in KGs denote every component in KG such as a unique identifier or literal value (e.g., string and integer) indicating entities and predicates

(e.g., relation type) placed between subject and object in triple. KG consists of the set of each resource E , R , and T^R , where E denotes the set of entities, R denotes the set of relation types, and T^R indicates the set of relation triples.

Triples. Triples in the set of relation triples T^R indicate the relationships of real-world objects or events following the format of $\langle \text{subject, predicate, object} \rangle$. Subjects and objects are from the set of entities E , and the predicates are selected for the set of relations R . For example, $\langle \text{Barack_Obama, IsChildOf, Barack_Obama_Sr.} \rangle$ is a relation triple where it denotes the relationship that Barack Obama Sr. is the child of the former president Barack Obama with a given triple format.

Triangular Structure. One of the most representative triangular structures is a cycle structure. Specifically, such as a set of relation triples $\langle e_1, \text{IsFriendOf}, e_2 \rangle$, $\langle e_2, \text{IsFriendOf}, e_3 \rangle$, and $\langle e_3, \text{IsFriendOf}, e_1 \rangle$ can form a cyclic structure. It is crucial for KG embedding models to be able to express such intricate relation structures explicitly to generate meaningful embeddings.

Relation Patterns. There are three main relation patterns frequently observed in KGs: symmetry, inversion, and composition. More precisely, symmetry can be a pattern such as $\langle e_1, \text{IsFriendOf}, e_2 \rangle$ and $\langle e_2, \text{IsFriendOf}, e_1 \rangle$, inversion can be a pattern between $\langle e_1, \text{IsStudentOf}, e_2 \rangle$ and $\langle e_2, \text{IsTeacherOf}, e_1 \rangle$, and lastly, an example of composition can be a pattern where the $\langle e_1, \text{IsFatherOf}, e_3 \rangle$ exists when $\langle e_1, \text{IsSpouseOf}, e_2 \rangle$ and $\langle e_2, \text{IsMotherOf}, e_3 \rangle$ hold.

3.2. Knowledge Graph Alignment. Formally, the problem of embedding-based KG entity alignment can be defined as follows.

Problem 1. With two different input KGs, $\text{KG}_s = (E_s, R_s, T_s^R)$ and $\text{KG}_t = (E_t, R_t, T_t^R)$ where each component denotes entities, relations, and relation triples in order, KG alignment techniques focus on computing distance between embeddings of two entities $v(e_i)$ and $v(e_j)$ where $e_i \in E_s$ and $e_j \in E_t$.

Given the distances between every entity pair from KG_s to KG_t , it identifies the corresponding entities under the premise that the entity with the shortest distance from the current entity embedding is the most likely to be equivalent to it.

Seed Alignment. Most existing KG entity alignment techniques train their model in a supervised or semisupervised manner which requires a sufficient amount of prealigned entities as their reference. These prealigned entities are often called as *seed alignment* and can be obtained with various approaches such as leveraging semantic owl:sameAs in DBpedia. However, seed alignments demand lots of human effort to acquire which makes the process costly resulting in a lack of seed alignment in real-world datasets.

Entity Similarity. The similarity between entities is decided by the distance of the embeddings in a point-wise space. The shorter distance denotes that the two entities are more likely to be the same entities. The final alignments are determined by the top-1 closest entity among other candidates. The distance can be calculated in the embedding space with various types of distance metrics. The most popular measures are Manhattan distance, L2 norm, and cosine similarity.

Importation notations are summarized in Table 1.

3.3. Requirements. In the scope of this work, there are three main requirements that need to be satisfied which are as follows:

(R1) *Relation Pattern-Aware Embedding.* To guarantee accurate embeddings, it is crucial to integrate relation pattern information. In order to capture the aforementioned four major relation patterns, it has been validated that the rotation constraint [16] can be applied in complex spaces. While making the embeddings to capture the semantic structure with relation patterns, we also need to consider alignment-oriented features to enforce the equivalent embeddings to be close as much as possible. The details of the implementation of this process are provided in Section 4.

(R2) *Neighborhood-Aware Complex Embedding.* Along with the semantic features of triples, we need to consider the similarity of neighborhood structure information of entities. Since a number of research studies have shown that the sophisticated calculus can be performed with embeddings in complex spaces, it is highly on demand to explore the GCNs that operate with such expressive computations. In the scope of this paper, we propose a novel GCN framework that generates complex embeddings by using the Hadamard product-inspired method. We discuss it in more detail in Section 5.

(R3) *Auxiliary Relation Information Integration.* The expressiveness and the accuracy of the embeddings can be further improved by integrating auxiliary relation information such as relation type and direction. However, the majority of the existing GCNs ignores the supplementary relation information and simply propagates the connectivity information. Besides, integrating such additional information often requires excessive parameters and increases the complexity of the computation. To overcome such problems, we design simple but effective approaches to integrate relation type and direction information. A more in-depth explanation about this approach is provided in Section 5.

3.4. Framework Overview. To realize the existing challenges, we designed a novel framework, namely, ComplexGCN, consisting of two main modules: rotation-based embedding module (RE) and deep integration (DI) module. As

TABLE 1: Overview of important notations.

Notation	Description
KG_1, KG_2	Input source and target knowledge graphs
Img, Re	Imaginary axis and real axis
e_h, e_t	Head entity and tail entity from triple
$d(e_i, e_j)$	Distance between entity e_i and entity e_j
$\ e_h \circ h - e_t\ $	Rotation-based score function
$h_v^{l-1}, h_u^{l-1}, h_u^{l-1}$	Hidden representations of node v and its neighbors at $(l-1)$ -th layer
$\text{rt}(\cdot)$	Hadamard product-based composition operator
\rightarrow, \leftarrow	Outgoing and incoming directions of relation
$h_{v_r}^l, h_{v_{im}}^l$	Real and imaginary parts of a node embedding at l -th layer
W^l, W^l_r	Weight and relation weight matrices at the l -th layer
$S_e^{\text{RE}}, S_e^{\text{DI}}$	Similarity results from RE and DI modules
S_e	Final similarity result

demonstrated in Figure 3, there are two main components in our framework: *rotation-based embedding module* in Section 4 and *deep integration module* in Section 5. The overview of our framework is presented as follows:

Rotation-Based Embedding Module. First, to be able to capture the four prevalent relation patterns, we exploit the rotation score function [16] which rotates the head entity with the constrained relation embedding and minimizes the distance between the rotated head entity and tail entity in each triple in complex spaces. By rotating the entities by given angles instead of adding them [9], the relation patterns can be aptly captured by solving contradictions that the translation assumption often neglects (R1). Note that, this module is different from the original RotatE in the sense that we enforce the embeddings to secure both semantic and alignment information by optimizing the semantic-preserving objective function and alignment loss function at the same time.

Deep Integration Module. Despite the promising direction of the RE module, there still remains space for improvement in terms of neighborhood information. Neighboring entities and the relations that connect the entity with its neighbors play an essential role in the alignment task since there is a tendency that the entities on par to have a similar neighborhood. In our work, we propose a deep integration module (DE) which is a generalized GCN framework that is inspired by reference [19] to generate complex embeddings while incorporating neighboring information (R2) with extra relational information. Specifically, the DE module captures the direction and type information of relations with specific weight matrices and rotation composition operations (R3). Besides, our DE module enables the Hadamard product-inspired computation between direction-specific weight matrices and hidden node embeddings to give more expressiveness through a more sophisticated calculus (R2).

4. Rotation-Based Embedding Module

This section first discusses the learning approach of the semantic-preserving embeddings based on the rotation

assumption which was devised by the authors in reference [16]. We then train them to retain both semantic and alignment features by constraining with two individual loss functions.

4.1. Rotation-Based Embedding. The ultimate goal of the RE module is to learn alignment-oriented complex vector representation of entities and relations by preserving relation triple information. Hence, we leverage the rotation function [16] to generate complex embeddings that better reflect complicated relational structures and relation patterns in complex space \mathbb{C}^n . To estimate the legitimacy of each relation triple $t \in T^R$, we calculate the rotation-based score function as

$$f_r(t) = \|\mathbf{c}(h) \odot \mathbf{c}(r) - \mathbf{c}(t)\|, \quad (1)$$

where $\mathbf{c}(\cdot)$ denotes complex vector representation, \odot indicates entrywise product, $\|\cdot\|$ is the Manhattan distance metric, and h, r , and t denote head entity, relation, and tail entity, respectively. In essence, the score function is based on Euler's formula $e^{i\theta} = \cos(\theta) + i \sin(\theta)$ which is another form of the complex number $z = a + bi$. By constraining the modulus of every relation representation $|e^{i\theta_{r,i}}|$ ($r = e^{i\theta_{r,i}}$) to be equal to 1, entrywise multiplication between head entity embedding and relation embedding $e^{i\theta_{r,i}}$ can be regarded as rotation from head entity to tail entity by $\theta_{r,i}$ radians. For example, if we assume $\mathbf{c}(r)$ is $i = e^{i90^\circ}$, then the multiplication of $\mathbf{c}(h) = e^{i\theta_{e,i}}$ and $\mathbf{c}(r)$ equals to $e^{i(90^\circ+\theta)}$. However, the rotation-based score function is not tailored for the alignment problems since the purpose of such a score function is to recognise the semantics in triples. Hence, we further combine two different loss functions to ensure that the RE module is alignment-oriented.

It is worth noting that such rotation calculus in complex spaces guarantees that the model is able to capture three popular relation patterns along with the cycle structure. The following definitions demonstrate the effectiveness of the rotation function in preserving the relation patterns and structure.

Definition 2. A relation r_i is symmetric, if and only if the phase $\theta_{r,i}$ of relation embedding $e^{i\theta_{r,i}}$ equals to 0 or π .

5.1. Complex Graph Convolutional Networks. To harness the advantage of hypercomplex space, QGNN [19] has first introduced a GNN model that can produce embeddings within the quaternion space. Borrowing the computation method of the Hamilton product, the product between the weight matrix and the hidden node vector representation has been defined as follows:

$$h_v^{(l+1),Q} = \sigma \left(\sum_{u \in \mathcal{N}_v \cup \{v\}} a_{v,u} W^{(l),Q} \otimes h_u^{(l),Q} \right), \quad (9)$$

where σ is the activation function, \mathcal{N}_v^r is the set of neighbors of v , \otimes denotes the Hamilton product, $h_v^{(l+1),Q}$ is the quaternion embedding of node v in $(l+1)$ th layer, $a_{v,u}$ denotes the edge constant, and $W^{(l),Q}$ is the weight matrix. Since QGNN leverages the weight-sharing method across the real and imaginary component vectors of a node representation, it enables capturing the subtle differences between different nodes. To the best of our knowledge, such a GCN framework is compatible with complex spaces that have not been handled in our research field so far. Despite the achievement of QGNN, it is still worth developing a generalized GCN framework that works in complex spaces since embeddings in complex spaces still show the runner-up performance in practice while having less degree of freedom.

5.1.1. Direction-Aware Hadamard Product. To reflect direction information, we adopt two types of direction-aware weight matrices W_{d-} and W_{d+} for incoming and outgoing neighbors, respectively. We use individual weight matrix W_s which is specified for self-loop to integrate the embedding of the source entity from the previous layer with a relation that connects it to itself. Moreover, we aim to replace the Hamilton product with the Hadamard product for the following reasons. First, the former is applicable to square matrices, while the latter can be applied to matrices of any size. This is suitable for imbalanced graph alignment, where the two input graphs do not have the same size. Second, the Hadamard product is straightforward to compute, as it requires only elementwise multiplications.

More precisely, going beyond QGNN [19], we reconstruct the weight matrix as a nodewise concatenation of the direction-aware matrices and the individual matrix, that is, $W = \langle W_{d-}, W_{d+}, W_s \rangle$. In that, each inside matrix is defined as follows:

$$\begin{cases} W_{d+} = W_{d+r} + W_{d+i} \mathbf{i}, \\ W_{d-} = W_{d-r} + W_{d-i} \mathbf{i}, \\ W_s = W_{s^r} + W_{s^i} \mathbf{i}, \end{cases} \quad (10)$$

where $W_{\cdot r}$ denotes the real part of the corresponding weight matrix and $W_{\cdot i} \mathbf{i}$ is the imaginary part of the weight matrix. Each $+$, $-$, and s denotes outgoing directions, incoming directions, and self-loop of relations. Unlike R-GCNs [15] where a specific weight matrix is assigned to each relation type, we leverage the weight-sharing mechanism for the three aforementioned relation directions so that the model can avoid the overparameterization problem. While keeping

the simplicity of the model, it still can capture meaningful features by incorporating relation-type information with the rotation-based composition operator.

The new version of the weight matrix allows the Hadamard product between the weight matrix and the node representation to successfully generate complex embeddings with richer information. More precisely, the Hamilton product in equation (9) is replaced by the Hadamard product between the abovementioned weight matrices and the hidden complex representation of entities that can be described as follows:

$$W_r^{(l)} h_{v^r}^{(l)} + W_i^{(l)} h_{v^i}^{(l)} \mathbf{i} - W_i^{(l)} h_{v^r}^{(l)} + W_r^{(l)} h_{v^i}^{(l)} \mathbf{i}. \quad (11)$$

For the sake of simplicity, the equation can be expressed as matrix multiplication represented as

$$W^{(l)} \odot h_v^{(l)} = \begin{bmatrix} 1 \\ i \end{bmatrix}^\top \begin{bmatrix} W_r^{(l)} & W_i^{(l)} \\ -W_i^{(l)} & W_r^{(l)} \end{bmatrix} \begin{bmatrix} h_{v^r}^{(l)} \\ h_{v^i}^{(l)} \end{bmatrix}, \quad (12)$$

where W_r is the real part of the weight matrix, W_i denotes the imaginary part of the weight matrix, h_v^r is the real part, and h_v^i is the imaginary part of the hidden node embedding. By transforming the hidden representations in complex space instead of the Euclidean space through the Hadamard product, we can more rigorously tune the embeddings which can result in capturing the features more precisely.

After propagating through the GCN framework, the final vector representations are generated by concatenating the hidden node representations at all GCN layers and then performing the linear transformation on them. Hence, we can incorporate local and global aspects of the neighborhood information to make the embeddings more consolidated.

5.1.2. Entity-Relation Composition. The key idea to incorporating relational-type information is to feed forward the synthesis of entity and relation through the GCN layers. To keep its expressiveness and simplicity in complex spaces, we leverage the rotation concept using the Hadamard product as follows:

$$\text{rt}(h_u^{(l)}, h_{vur}^{(l)}), \quad (13)$$

where $\text{rt}(\mathbf{x}, \mathbf{y})$ denotes the Hadamard product between \mathbf{x} and \mathbf{y} , $h_u^{(l)}$ represents the entity embedding, and $h_{vur}^{(l)}$ denotes the relation embedding that connects entity u and v . Note that, to follow the track of relation information, the hidden embedding $h_{v^i}^{(l)}$ in equation (12) is replaced by equation (13).

5.1.3. Put It Altogether. In the end, we design a novel GCN mechanism that learns complex embeddings in a more careful manner while considering both relation type and direction information at the same time via end-to-end learning. As the embeddings in the complex plane (i.e., the Argand plane) can imply meaningful semantic information through the entrywise product (i.e., the Hadamard product), we also leverage this property into our GCN model to move beyond the traditional multiplication between weight matrix

and embeddings. More precisely, the embeddings of entities are updated via the following propagation rule per node:

$$h_v^{(l)} = \sigma \left(\sum_{(u,r) \in \mathcal{N}(v)} W_d^{(l)} \odot \mathbf{tr}(h_u^{(l-1)}, h_{\text{vr}}^{(l-1)}) + W_s^{(l)} \odot \mathbf{rt}(h_v^{(l-1)}, h_{\text{vr}}^{(l-1)}) \right), \quad (14)$$

where σ denotes the activation function, $\mathcal{N}(\cdot)$ denotes the set of pair of neighboring nodes and corresponding relations, $W_d = \langle W_{d-}, W_{d-} \rangle$, and W_s are trainable weight matrices, $h_v^{(l)}$ is the hidden representation of node v in layer l , $\mathbf{tr}(\cdot)$ is the Hadamard product-based composition operator inspired by rotate [16], and \odot denotes the Hadamard product between the weight matrix and node representation. Note that, the reasons W_d and W_s are separated instead of combined into one weight matrix as the former represents neighbor information, while the latter preserves the node identity across layers.

In summary, we use a different approach compared to QGNN to incorporate the information of relation types by updating embeddings not only with its neighbors but also with the neighbors and their corresponding relations that connect them to the source entity.

5.2. Loss Function. To enforce aligned entities to be closer than those that are not, we utilize prealigned entities set \mathbb{L}_e as a reference when training the DI module. The model aims to minimize the below margin-based scoring function [13].

$$\mathcal{O}_{\text{de}} = \frac{1}{\|\mathbb{L}_e\|} \sum_{(e_i, e_j) \in \mathbb{L}_e} \sum_{(e_i, e'_j) \in \mathbb{L}'_e} \max\{0, d(\mathbf{v}(e_i), \mathbf{v}(e_j)) - d(\mathbf{v}(e_i), \mathbf{v}(e'_j)) + \gamma_a\}, \quad (15)$$

where γ represents the hyperparameter, \mathbb{L}'_e is a set of corrupted entity pairs, and $d(\cdot)$ denotes the cosine distance. \mathbb{L}'_e is built by replacing either e_i or e_j in truly alignment entity pairs with an entity in s -nearest neighbor candidates [13]. By sampling the negative alignments with close neighbors instead of randomly selecting from a uniform distribution, it can enforce the model to be able to distinguish subtle differences between a truly corresponding counterpart and the falsely corresponding one which has a high alignment possibility.

It is noteworthy that the real and the imaginary parts of the hidden node embeddings are obtained via the backward propagation when training the model with the loss function in equation (15). In particular, $h_{v,r}$ and $h_{v,i}$ are treated as two variables in equation (11); thus, they can be obtained using an optimizer such as stochastic gradient descent. The partition of hidden embeddings into real and imaginary parts enables the representation learning of high-order features, thus capturing the subtle difference between correct and incorrect alignments.

5.3. Alignment Inference. The final alignment score is determined by considering both similarity results generated from RE and DI modules. To effectively integrate two individual similarity results, it is efficient to utilize the late-fusion method instead of the early-fusion approach to maintain the characteristics of separated embedding spaces. Before we calculate the final result, we concatenate the embeddings generated from each GCN layer in order to consider from local range to the global scope of neighborhood information. Then, we fuse \mathbf{S}^{RE} and \mathbf{S}^{DI} by following the late-fusion method as follows:

$$\mathbf{S}_e = \beta \mathbf{S}_e^{\text{RE}} + (1 - \beta) \mathbf{S}_e^{\text{DI}}, \quad (16)$$

where $\beta \in [0, 1]$ denotes the balance coefficient that defines the importance of each module. Note that, in our model, similarity scores are calculated by measuring the cosine distance between entities in every entity pair. Finally, the likely-to-be-aligned counterparts are selected by the popular heuristic approach such as the greedy search algorithm. To retrieve the one-to-one alignments, we first sort the possible candidates of each entity in a descending order of similarity score and choose the one with the highest cosine similarity $\mathcal{S}(i, j)$ as the final alignment. Then, the entries in the similarity matrix which represent the similarity between i or j and other nodes are deleted. With the updated similarity matrix, we repeat the same procedure and end the iterations when all the entities of one of the source and target KGs have been matched.

6. Experiments

6.1. Experiment Setup

6.1.1. Datasets. We conduct experiments on the eight benchmark datasets constructed by the authors in reference [23] using the IDS (iterative degree-based sampling) algorithm. Based on the renowned real-world KGs: DBpedia [1], Wikidata [24], and YAGO [25], we adopt two monolingual and three multilingual KG datasets with tens of thousands of entities to evaluate the performance in various cases. The chosen curated datasets are divided twofold depending on the density of the average node degree: (1) V1: sparse version and (2) V2: denser version. V2 has a more similar structure to the real-world dataset by keeping the nodes with a high

degree while pruning those that are connected to fewer neighbors. The detailed statistics of the V1 and V2 datasets are organized in tables Tables 2 and 3.

6.1.2. *Metrics.* We utilize three representative metrics that are widely used for assessing the performance of KG alignment techniques.

- (i) Hits@m (the higher the better): Hits@m is one of the widely adopted measures that aims to estimate the alignment performance based on whether the true positive alignment appears within the top-m ranked candidates.

$$\text{Hits}@m = \frac{\sum_{v_s \in V_s} \mathbf{1}_{S(v_s, v_t) \in \text{top-}mS(v_s)}}{\#\{\text{True anchor links}\}}. \quad (17)$$

- (ii) MR (mean rank, the lower the better): to examine the alignment in terms of how high the anchor node is ranked among other anchor candidates, we use MR (mean rank) formulated as follows:

$$\text{MR} = \text{mean}(\text{ra}), \quad (18)$$

where ra indicates the rank of a truly aligned target node among the other candidates.

- (iii) MRR (mean reciprocal rank, the higher the better): another popular metric is MRR, represented as follows:

$$\text{MRR} = \text{mean}\left(\frac{1}{\text{ra}}\right), \quad (19)$$

where alignment performance is measured based on the average of reciprocal rank of truly aligned entities.

6.1.3. *Comparative Models.* To investigate the ability of our model, we carefully choose 7 competitive state-of-the-art KG alignment approaches organized as follows:

- (i) *MTransE.* It is based on the translation assumption to train the alignment-oriented embeddings of entities and relations and then learns the transformation matrix to assist truly matching pairs to have similar embeddings [8]
- (ii) *JAPE.* It utilizes both relation and attributes triples to generate embeddings and uses the parameter-sharing method to unify the embedding spaces [6]
- (iii) *GCN-Align.* It employs GCN (graph convolutional networks) to produce entity embeddings while considering both relation and attribute triples simultaneously [14]
- (iv) *BootEA.* It augments the training data during iterations by labeling close embeddings as newly generated seed alignments by using the bootstrapping method [7]
- (v) *RDGCN.* It generates relation-aware embeddings by encouraging attentive interaction between input KGs and their dual relation graphs. Then, it feed-

TABLE 2: Statistics of V1 datasets.

V1 datasets		#Ent	#Rel	#Rel.tri
EN-FR-15K-V1	English	15,000	267	47,334
	French	15,000	210	40,864
EN-DE-15K-V1	English	15,000	215	47,676
	German	15,000	131	50,419
DBP-15K-V1	English	19,993	1,178	97,253
	French	19,661	886	98,566
D-W-15K-V1	DBpedia	15,000	248	38,265
	Wikidata	15,000	169	42,746
D-Y-15K-V1	DBpedia	15,000	165	30,291
	YAGO3	15,000	28	26,638

TABLE 3: Statistics of V2 datasets.

V2 datasets		#Ent	#Rel	#Rel.tri
EN-FR-15K-V2	English	15,000	193	96,318
	French	15,000	166	80,112
EN-DE-15K-V2	English	15,000	169	84,867
	German	15,000	96	92,632
DBP-15K-V1	English	19,993	1,235	135,297
	French	19,661	1,126	128,493
D-W-15K-V2	DBpedia	15,000	167	73,983
	Wikidata	15,000	121	83,365
D-Y-15K-V2	English	15,000	72	68,063
	French	15,000	21	60,970

forwards the embeddings through two-layer GCNs with highway gates [13]

- (vi) *AliNet.* It extends the scale of the neighborhood to incorporate informative distant neighbors and then aggregates neighboring nodes by giving high importance to helpful neighbors based on attention GNNs and gating mechanism [26]
- (vii) *Multike.* It generates three perspectives of entity features (entity name, relation, and attribute) and then incorporates them with three combination strategies to consider multiple views of alignment information [27]
- (viii) *Dual-AMN.* It captures both intragraph and cross-graph information while minimizing computational complexity with a method for selecting challenging negative samples with reduced loss impact [28]

Note that, there are recent graph embedding methods such as QGNN [19] and magnet [20]. However, they mostly focus on single-graph tasks such as node classification and link prediction, and thus, they are not applicable to our setting. We include their evaluations in subsection 6.3 for reference.

6.1.4. *Training Details.* Following reference [23], we set the ratio of the seed alignments for training as 20 %, validation as 10 %, and test as 70 % for all datasets. To search for the suitable hyperparameters, we take the listed values as candidates: learning rate in $\{1e^{-4}, 5e^{-4}, 1e^{-3}, 5e^{-3}, 1e^{-2}, 5e^{-2},$

$1e^{-1}, 5e^{-1}$ }, margin γ_s and γ_a in $\{1, 2, 5\}$, matrix balancing weight β in $\{0.4, 0.5, 0.6\}$, loss balancing weight α in $\{10, 20, 50\}$, number of GCN layers in $\{1, 2, 3\}$, and embedding dimension size in $\{100, 200, 300\}$. The final selected hyperparameters are learning rate = $5e^{-4}$, $\gamma_s = 5$, $\gamma_a = 1$, $\beta = 0.6$, $\alpha = 50$, number of GCN layers = 2, and embedding dimension size for RE module = 100 and for DI module = 300. In addition, we set the maximum epoch number as 2000 and check the MRR result at every 10 epochs on the validation set and terminate the training before it hits the maximum epoch when MRR starts to decrease in succession. Table 4 summarizes the hyperparameters.

6.1.5. Implementation. We leveraged the PyTorch library along with Python 3.6 to implement and run our model. To execute the experiments in a fair manner, we ran the experiment over 100 times for each technique. Experiments were conducted on an AMD Ryzen Threadripper 3.8 GHz system with 64 GB RAM and four GTX Titan X graphic cards. We used random initializations.

6.2. End-to-End Comparison. We conduct an end-to-end comparison between ComplexGCN and selected state-of-the-art baseline techniques on real-world monolingual and multilingual KG datasets.

6.2.1. Monolingual Entity Alignment. The objective of the monolingual entity alignment task is to find out matching entities across two different input KGs constructed in the same language (i.e., English). For the evaluation metrics, we take four measures into consideration: Hits@1, Hits@10, MR, and MRR. The alignment technique is regarded as efficient with a higher Hits@k score and MRR, while a lower MR indicates high performance. Table 5 provides convincing results for our experiment. As can be seen from Table 5, our model consistently outperforms all other baseline techniques on four monolingual datasets. Specifically, ComplexGCN achieves up to 97.5% on Hits@1, 99.4% on Hits@10, and 0.983 on MRR.

From the experiment result, we can rationalize the effectiveness of two main properties of our model: (1) the Hits@1 and MRR of the result of the Dual-AMN, RDGCN, and GCN-Align demonstrate that incorporating multi-perspective relation information assists the performance gain. From this, it can be found that our rotation-based composition operator in the DI module makes our model efficient. (2) The large gap in accuracy between MTransE and ComplexGCN is observed in all scenarios, which implies that our DI module takes an important role in alignment performance improvement.

Following ComplexGCN, both BootEA and RDGCN and Dual-AMN have the highest results among the baseline techniques, while MTransE obtained the lowest output among all the other baseline techniques. This indicates that more complicated architecture often achieves better performance than solely relying on the translation constraint method. Moreover, GNN-based techniques such as GCN-

Align, RDGCN, AliNet, and Dual-AMN consistently exhibited promising results which indicate the significance of GNNs on the KG alignment task.

6.2.2. Multilingual Entity Alignment. In practice, there often exists a huge knowledge gap between different language-version KGs. It is crucial to discover equivalent entities between KGs in different languages; however, such a task requires a more sophisticated approach to address. The same alignment metrics are adopted to estimate the performance of the techniques on multilingual datasets for the monolingual test. From Table 6, the results that are similar to the experiment on monolingual datasets can be observed.

Different from the monolingual test, Dual-AMN, RDGCN, and MultiKE were the runner-up techniques among the baselines in most of the scenarios. It can be assumed that the algorithms that involve various pieces of information about KG are more suitable for multilingual settings. In addition, the margin between the Hits@1 results of our model and the runner-up technique was larger than those in a monolingual environment. Such a large margin of up to 12.9% on Hits@1 demonstrates that our model performs better on multilingual datasets, which validates the robustness of the relation-aware method as well as the power of the architecture of our model on multilingual datasets.

6.2.3. Running Time. Time complexity is one of the important metrics that evaluate the performance of the models since the shorter the running time, the more practical they are at the application level. As depicted in Figure 4, the running time of our model is quite competent in the D-Y-V2, D-Y-V1, D-W-V2, and D-W-V1 datasets (BootEA is the slowest method). However, our model is much slower in EN-FR-V2, EN-FR-V1, EN-DE-V2, EN-DE-V1, DBP-V2, and DBP-V1 datasets. This could be explained by the fact that two different languages have inherently different semantics that could not be aligned perfectly, whereas our rotation-based embedding tries to find the perfect entity calibration, which leads to slow convergence.

There are possible improvements to reduce run time such as simplifying tensor products with circular correlation operation [29] and NeuralLP [30]. However, these methods are often applied to more than one million entities, which is out of the scope of our datasets. Thus, they will be considered as our orthogonal future work.

6.3. Ablation Test. To analyse the effectiveness of each module in our model, we conduct the ablation study on the V1 and V2 D-W-15K datasets. We compare the results of our model and three variants as follows:

- (i) *Var-1.* In this variant, we discard the DI module and only keep the SE module, in order to evaluate the impact of the DI module
- (ii) *Var-2.* In this variant, we discard the SE module and only keep the DI module to see the impact of the SE module

TABLE 4: Hyperparameter configuration.

Max epoch	2000							
eval_freq	10							
Learning rate	$1e^{-4}$	$5e^{-4}$	$1e^{-3}$	$5e^{-3}$	$1e^{-2}$	$5e^{-2}$	$1e^{-1}$	$5e^{-1}$
emb_dim	100							
Margin γ	1							
GNN layer	1							
Negative sample	10							
Balancing weight α	10							
Balancing weight β	0.4							

TABLE 5: Experiment results on monolingual datasets.

Datasets	Metric	MTransE	JAPE	GCN-Align	BootEA	RDGCN	AliNet	MultiKE	Dual-AMN	ComplexGCN
D-W-15K-V1	Hits@1	0.258	0.251	0.362	0.574	0.517	0.472	0.410	<i>0.628</i>	0.683
	Hits@10	0.543	0.544	0.647	0.794	0.717	0.704	0.582	<i>0.816</i>	0.863
	MR	331.2	243.6	765.3	286.5	508.5	575.8	275.6	267.4	93.9
	MRR	0.354	0.349	0.463	0.648	0.582	0.553	0.466	<i>0.673</i>	0.749
D-W-15K-V2	Hits@1	0.272	0.263	0.507	0.823	0.621	0.741	0.497	0.794	<i>0.802</i>
	Hits@10	0.585	0.581	0.819	0.950	0.807	0.926	0.726	<i>0.953</i>	0.960
	MR	146.1	99.1	146.1	18.5	229.1	72.4	38.7	9.6	6.9
	MRR	0.374	0.369	0.611	0.865	0.682	0.806	0.571	0.833	<i>0.858</i>
D-Y-15K-V1	Hits@1	0.464	0.472	0.464	0.737	<i>0.930</i>	0.571	0.903	0.861	0.975
	Hits@10	0.732	0.744	0.663	0.874	<i>0.971</i>	0.725	0.951	0.964	0.994
	MR	245.6	211.3	1113.5	365.1	<i>17.9</i>	532.7	19.6	18.1	4.7
	MRR	0.557	0.567	0.537	0.788	<i>0.949</i>	0.631	0.923	0.895	0.983
D-Y-15K-V2	Hits@1	0.444	0.945	0.875	<i>0.959</i>	0.936	0.951	0.857	0.887	0.970
	Hits@10	0.705	0.626	0.964	0.991	0.973	<i>0.992</i>	0.925	0.979	0.994
	MR	85.1	82.4	47.2	4.9	13.9	6.1	10.1	5.2	1.4
	MRR	0.534	0.441	0.906	0.966	0.950	<i>0.967</i>	0.881	0.921	0.979

The results with italics denote runner-ups. Bold values mean best performance.

- (iii) *Var-3*. In this variant, we replace the DI module with vanilla GCN to validate the effectiveness of our relation-aware DI module
- (iv) *Var-4*. In this variant, we use a single weight matrix instead of three as in equation (10). This is actually a better version of QGNN [19] that leverages the design of our DI module
- (v) *Var-5*. In this variant, we replace the DI module with a magnet [20] to validate the effectiveness of our relation-aware DI module.

As it is demonstrated in Table 7, the original version of our model outperforms other variants. More precisely, the margin of at least 45% on Hits@1 is observed between Var-1 and the original ComplexGCN which indicates that our DI module significantly contributes to performance improvement. Second, another variant Var-2 also shows lower accuracy which justifies that it is an effective approach to take both SE and DI modules into consideration. This shows that both semantic and neighborhood information is worth incorporating to enhance the quality of the alignment-powered embeddings. Third, the comparison between Var-3 and our model proves the comparative superiority of our newly proposed GCN architecture compared to vanilla GCN. According to the results, our GCN framework ranks the truly aligned entities higher than vanilla GCN overall which can evidence that our proposal works better on the

entire entities evenly. We assume that this is because of the capability to capture various types of relation patterns. In other words, our model can generate more accurate embeddings for the entities that are connected with specific relation patterns along with general relations. Fourth, the variant Var-4 is close to our performance by being the runner-up in some cases. This is reasonable as it is almost equivalent to QGNN. However, it does not separate the weight matrix to capture the direction information and self-loop information, thus losing some contexts for the graph alignment setting. Finally, the variant Var-5 is worse than our original ComplexGCN, even though it leverages a recent graph embedding process for complex spaces. It could be explained by the fact that it was originally designed for node classification and link prediction only.

6.4. Supervision Level Study. The ability to achieve high performance with a small amount of prealigned entities is a crucial challenge for KG alignment techniques. We trained our model and the other baseline techniques with different ratios of training data to see their degree of performance degradation depending on the amount of the seed alignment. From Figure 5, our model maintains the highest result or the same level of accuracy with all the baselines even under the condition where at least 3% of seed alignments are given. Other datasets are omitted for brevity's sake because

TABLE 6: Experiment results on multilingual datasets.

Datasets	Metric	MTransE	JAPE	GCN-Align	BootEA	RDGCN	AliNet	MultiKE	Dual-AMN	ComplexGCN
EN-FR-15K-V1	Hits@1	0.247	0.261	0.336	0.508	<i>0.754</i>	0.388	0.747	0.727	0.883
	Hits@10	0.563	0.595	0.681	0.793	0.881	0.829	0.843	0.940	0.976
	MR	251.7	175.4	562.1	227.6	156.1	483.2	97.8	68.1	13.3
	MRR	0.352	0.372	0.451	0.600	0.801	0.485	0.781	0.802	0.915
EN-FR-15K-V2	Hits@1	0.241	0.292	0.415	0.660	0.847	0.581	<i>0.864</i>	0.842	0.968
	Hits@10	0.240	0.623	0.796	0.904	<i>0.934</i>	0.878	0.923	0.855	0.996
	MR	206.2	89.1	131.5	131.5	61.7	94.0	12.4	74.9	1.9
	MRR	0.337	0.401	0.541	0.741	0.880	0.692	0.884	0.910	0.980
EN-DE-15K-V1	Hits@1	0.308	0.289	0.479	0.676	0.830	0.610	0.757	<i>0.841</i>	0.933
	Hits@10	0.611	0.609	0.754	0.863	<i>0.914</i>	0.831	0.826	0.896	0.984
	MR	224.1	140.5	352.3	125.4	67.3	216.5	91.6	72.8	12.8
	MRR	0.410	0.395	0.572	0.741	0.856	0.682	0.781	0.897	0.952
EN-DE-15K-V2	Hits@1	0.194	0.168	0.533	0.835	0.834	0.815	0.754	<i>0.853</i>	0.949
	Hits@10	0.432	0.417	0.781	0.935	<i>0.936</i>	0.930	0.834	0.898	0.990
	MR	193.4	139.9	108.1	16.1	74.8	71.3	45.1	21.3	3.7
	MRR	0.274	0.250	0.619	0.867	0.861	0.856	0.784	0.905	0.965
DBP-15K-V1	Hits@1	0.275	0.357	0.408	0.592	0.796	0.501	0.725	<i>0.855</i>	0.908
	Hits@10	0.578	0.602	0.718	0.828	0.894	0.803	0.834	0.971	0.980
	MR	235.7	156.3	433.8	176.2	115.1	354.9	94.3	67.4	13.2
	MRR	0.381	0.384	0.512	0.671	0.829	0.584	0.762	0.898	0.934
DBP-15K-V2	Hits@1	0.245	0.329	0.415	0.657	0.876	0.696	0.783	<i>0.897</i>	0.959
	Hits@10	0.576	0.675	0.774	0.871	0.952	0.904	0.942	0.980	0.993
	MR	217.5	137.6	115.9	53.7	75.6	85.2	25.8	11.5	2.8
	MRR	0.366	0.445	0.532	0.734	0.907	0.774	0.846	0.929	0.973

The results with italics denote runner-ups.

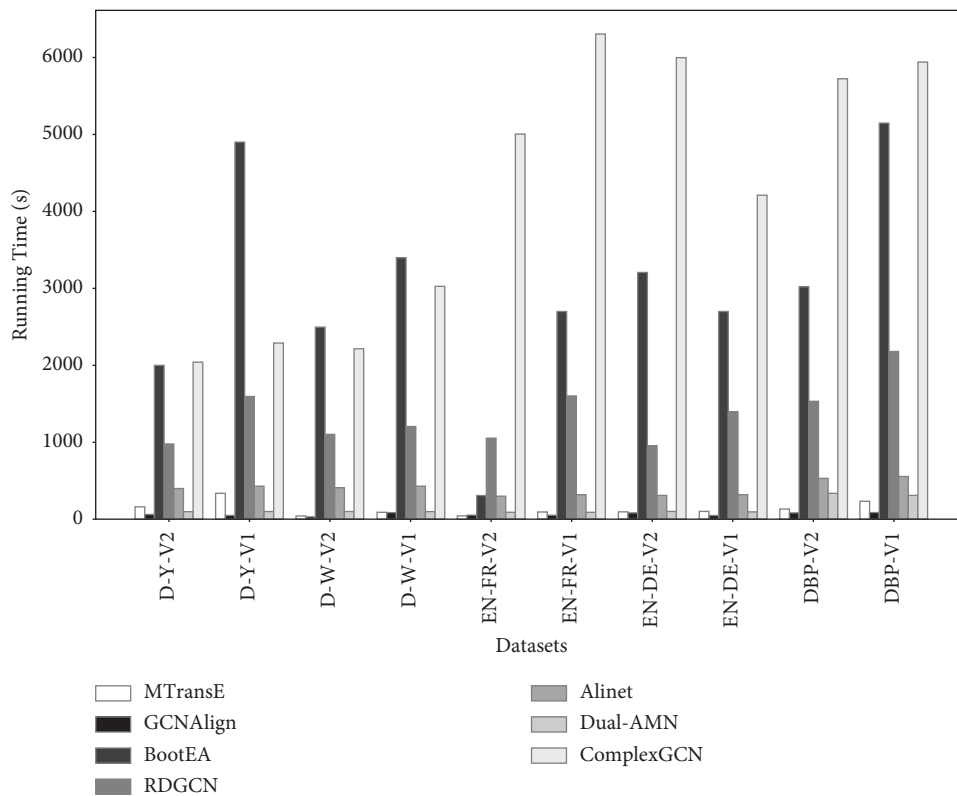


FIGURE 4: Time complexity in the log scale of the techniques on various datasets.

TABLE 7: Ablation test on D-W-15K-V1 and V2.

Datasets	Metric	Var-1	Var-2	Var-3	Var-4 (QGNN)	Var-4 (Magnet)	ComplexGCN
D-W-15K-V1	Hits@1	0.229	0.615	0.679	<i>0.680</i>	0.638	0.683
	Hits@10	0.468	0.791	<i>0.862</i>	0.859	0.826	0.863
	MR	584.9	174.5	106.9	<i>101.4</i>	104.3	93.9
	MRR	0.310	0.678	<i>0.746</i>	<i>0.746</i>	0.733	0.749
D-W-15K-V2	Hits@1	0.296	0.718	0.762	<i>0.771</i>	0.722	0.802
	Hits@10	0.662	0.893	<i>0.954</i>	0.947	0.932	0.960
	MR	60.7	75.2	11.1	9.2	17.4	6.9
	MRR	0.417	0.780	<i>0.825</i>	0.817	0.712	0.858

The results with italics denote runner-ups.

they show similar trends. This result demonstrates that our model can even preserve its performance in real-world practice where labeled data are hard to be accumulated.

7. Related Works

7.1. Embedding-Based KG Completion. In the past few years, extensive embeddings have been performed to handle the *incompleteness* problem of KGs. The major idea in this field is to effectively encode embeddings of resources of a single KG by capturing the plausibility of the relation triples in KGs so that the knowledge semantics can be expressed in Euclidean space. The generated embeddings are then leveraged to fill out the missing links (link prediction) or to find out which class of unlabeled entities should belong to (node classification). The most renowned approaches [31, 32] are based on the translation assumption which was first proposed by TransE [9]: $\text{head_entity} + \text{relation} \approx \text{tail_entity}$. The parameters are trained by inducing the score of positive triples to have lower values than those of the negative ones. Despite its simplicity and effectiveness, there remain several problems: (1) it fails to address the cyclic structure and (2) different embeddings are encoded for the same relation when it is symmetry. To mitigate these problems, vector representations where each entry is a complex number have emerged in the field [16, 17]. Unlike vectors of real numbers, complex vectors can harness the intuition behind Euler’s formula $e^{i\theta} = \cos(\theta) + i \sin(\theta)$ where $e^{i\theta}$ is a value at which the length of 1 bar is rotated from the real axis by θ radians. Taking advantage of this property, RotatE [16] constrains every relation embedding to have modulus 1 so that the relation embeddings can be deemed as rotation from head entity to tail entity. By bringing the rotation concept, all three relation patterns as well as cyclic structures can be easily captured.

7.2. Conventional KG Entity Alignment Methods. As aforementioned, before the era of representation learning of KGs started, *text-based EA approaches* were frequently adopted in many domains to enrich KGs with newly aligned entities in different KGs. Text-based EA approaches provide straightforward intuition to align entities in distinctive KGs under the assumption that equivalent entities have the same literal information such as entity name. The primary EA method appeared as the problem suited for relational data, mainly relying on the rich textual information (e.g., entity description) of entities in data. Most representative traditional

methods such as PARIS [33], RiMOM-IM [34], and LogMap [35] are tailored for finding matching between entities (instances) or even for relations (also known as schema) and classes as well. To detect alignments, text-based approaches for KGs exclusively employ attribute triples as their input where the source entity is connected to a textual value by an attribute type. In the case of cross-lingual KG matching, the use of machine translation (e.g., Google translate) is imperative to convert non-English-version KG into English for making them easily comparable. It is noteworthy that such conventional approaches are subject to a lack of a sufficient amount of textual information on entities in real-world KGs leading to requiring novel methods to take account into more conducive information for alignment tasks.

7.3. Embedding-Based KG Entity Alignment. Unlike conventional KG alignment methods, *embedding-based KG alignment approaches* have shown to be beneficial at leveraging multifaceted information and simplifying the inference process to compute matchable probability between every entity in source and target KGs. The core idea is to embed entities or relations into continuous vector space and measure the similarity between each vector point, so-called embedding, by measuring the distance between them with a given particular distance metric. Embedding-based KG alignment approaches can be categorized into two main groups: *translation-based models* and *neighborhood-integration models*. *Translation-based models* [6, 7] follow the concept of TransE [9] to conserve the knowledge logic of the input triples. Over the last few years, the interest in graph neural networks (GNN) has exponentially grown due to their capability of integrating neighborhood information. An entity can be expressed with the consolidations of its neighboring nodes since entities do not have any meaning without connection to other nodes while it can be defined with its connections to its neighbors. *Neighborhood-integration models* [13, 14, 26, 36–41] utilize GNN methods (e.g., recurrent GNN [42], graph convolutional network (GCN) [11], and graph attention network [12]) to incorporate neighborhood structure to infer alignment likelihood following the assumption that the equivalent entities in KGs tend to have similar neighborhood structure.

7.4. Embeddings in Complex Space. In recent literature, researchers have discovered that the entities encoded in complex or even hypercomplex spaces can have much

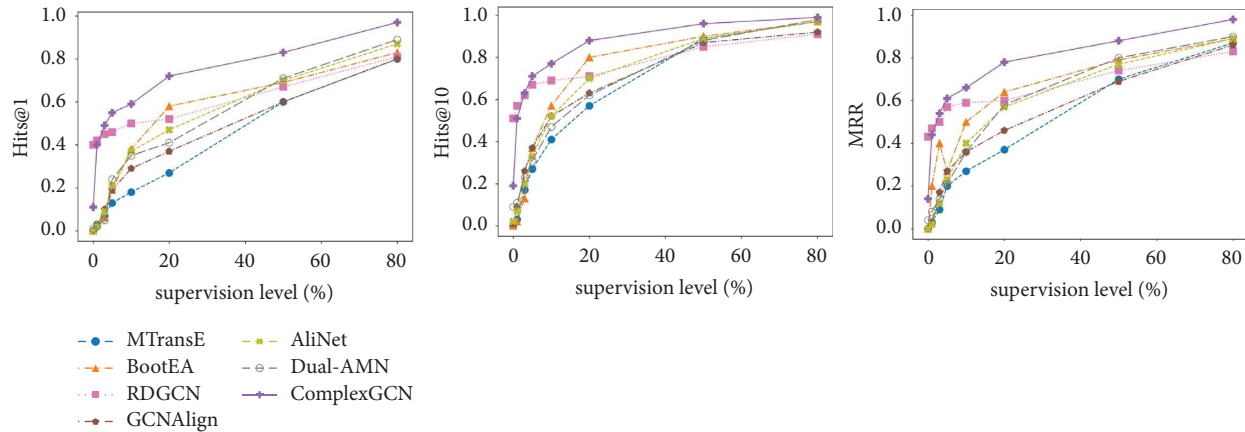


FIGURE 5: Alignment performance on the D-W-15K-V1 dataset with different supervision levels.

expressiveness in comparison with the embeddings represented with only real values. Formally, the pointwise representation of entities in complex space consists of a complex number which holds the form of $a + bi$ where a and b are real values and i denotes the imaginary unit. Each element in a complex vector follows the aforementioned form which allows the embeddings to have denser representation, hence enabling a variety of computations to operate such as rotation and conjugate [16–18]. Due to such properties, complex embedding learning methods have shown their promising performance which opens much potential to be further developed. ComplEx [17] is the first model to successfully capture symmetric and antisymmetric relation patterns. RotatE [16] further proposes a rotation-based score function to model additional composition patterns. There has been an effort to build a generalized GNN model in quaternion space, the so-called QGNN [19], by employing the Hamilton product in the GNN model. Similar to our effort is magnet [20], which introduces complex spaces for GNNs. However, magnet only focuses on node classification and link prediction; thus, it lacks important information for knowledge graph alignment. To the best of our knowledge, this is a pioneering paper in handling complex embeddings for graph alignment, especially knowledge graphs. Our novelty is a careful combination of a rotation-based embedding augmentation, a new loss function, and a new parametrization for weight matrices.

8. Conclusion and Future Works

This paper proposed a novel framework, namely, ComplexGCN that generates complex embeddings via rotation constraint and the advanced GNN module that performs in complex spaces. The model combines RE and DI modules to complement each other and guarantees the semantic and relation information-preserving embeddings which lead to more accurate alignment results. To this end, ComplexGCN first learns the semantic-aware complex node representations by constraining the triples with rotation assumption. Then, the model incorporates the multiperspective of relation information through the rotation composition operator and gains the expressiveness of the embeddings with

Hadamard product-inspired GNN architecture. Extensive experiments conducted on various datasets demonstrate the superiority of our model against the state-of-the-art baseline techniques and the comparison between the variants of our model justifies the design of our model. For future work, we plan to incorporate multimodal data and attribute triple information to further enhance the expressiveness of alignment-oriented embeddings.

Data Availability

The experiments were performed on public datasets (see Section 6.1.1).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Lehmann, R. Isele, M. Jakob et al., “Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia,” *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.
- [2] R. Speer and C. Havasi, “Conceptnet 5: a large semantic network for relational knowledge,” in *The People’s Web Meets NLP*, pp. 161–176, Springer, Berlin, Germany, 2013.
- [3] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song, “Variational reasoning for question answering with knowledge graph,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, February 2018.
- [4] Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.-K. Huang, and C. Xu, “Recurrent knowledge graph embedding for effective recommendation,” in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 297–305, Columbia, Canada, October 2018.
- [5] C. Wang, Y. Song, H. Li, M. Zhang, and J. Han, “Text classification with heterogeneous information network kernels,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AR, USA, February 2016.
- [6] Z. Sun, W. Hu, and C. Li, “Cross-lingual entity alignment via joint attribute-preserving embedding,” in *International Semantic Web Conference*, pp. 628–644, Springer, Berlin, Germany, 2017.

- [7] Z. Sun, W. Hu, Q. Zhang, and Y. Qu, "Bootstrapping entity alignment with knowledge graph embedding," *IJCAI*, vol. 18, pp. 4396–4402, 2018.
- [8] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual Knowledge Graph Embeddings for Cross-Lingual Knowledge Alignment," 2016, <https://arxiv.org/abs/1611.03954>.
- [9] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, vol. 26, 2013.
- [10] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: an efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 257–266, Anchorage, AK, USA, August 2019.
- [11] T. N. Kipf and M. Welling, "Semi-supervised Classification with Graph Convolutional Networks," 2016, <https://arxiv.org/abs/1609.02907>.
- [12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph Attention Networks," 2017.
- [13] Y. Wu, X. Liu, Y. Feng, Z. Wang, R. Yan, and D. Zhao, "Relation-aware Entity Alignment for Heterogeneous Knowledge Graphs," 2019, <https://arxiv.org/abs/1908.08210>.
- [14] Z. Wang, Q. Lv, X. Lan, and Y. Zhang, "Cross-lingual knowledge graph alignment via graph convolutional networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 349–357, Brussels, Belgium, November 2018.
- [15] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*, pp. 593–607, Springer, Berlin, Germany, 2018.
- [16] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge Graph Embedding by Relational Rotation in Complex Space," 2019, <https://arxiv.org/abs/1902.10197>.
- [17] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of the International Conference on Machine Learning*, pp. 2071–2080, PMLR, Baltimore, ML, USA, July 2016.
- [18] X. Huang, J. Tang, Z. Tan, W. Zeng, J. Wang, and X. Zhao, "Knowledge graph embedding by relational and entity rotation," *Knowledge-Based Systems*, vol. 229, Article ID 107310, 2021.
- [19] T. D. Nguyen and D. Phung, "Quaternion graph neural networks," in *Proceedings of the Asian Conference on Machine Learning*, pp. 236–251, PMLR, İstanbul, Turkey, November 2021.
- [20] X. Zhang, Y. He, N. Brugnone, M. Perlmutter, and M. Hirn, "Magnet: a neural network for directed graphs," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27003–27015, 2021.
- [21] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2022.
- [22] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [23] Z. Sun, Q. Zhang, W. Hu et al., "A Benchmarking Study of Embedding-Based Entity Alignment for Knowledge Graphs," 2020, <https://arxiv.org/abs/2003.07743>.
- [24] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [25] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum, "Yago: a multilingual knowledge base from wikipedia, wordnet, and geonames," in *International Semantic Web Conference*, pp. 177–185, Springer, 2016.
- [26] Z. Sun, C. Wang, W. Hu et al., "Knowledge graph alignment network with gated multi-hop neighborhood aggregation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 222–229, 2020.
- [27] W. Hu, Q. Zhang, Z. Sun, and J. Huang, *Multike: A Multi-View Knowledge Graph Embedding Framework for Entity Alignment*, ISWC, France, 2019.
- [28] X. Mao, W. Wang, Y. Wu, and M. Lan, "Boosting the speed of entity alignment 10x: dual attention matching network with normalized hard sample mining," in *Proceedings of the Web Conference*, pp. 821–832, Lyon, France, April 2021.
- [29] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Washington, DC, USA, December 2016.
- [30] F. Yang, Z. Yang, and W. W. Cohen, "Differentiable learning of logical rules for knowledge base reasoning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [31] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*, Austin, TX, USA, January 2015.
- [32] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Washington, DC, USA, February 2014.
- [33] F. M. Suchanek, S. Abiteboul, and P. Senellart, "Paris: Probabilistic Alignment of Relations, Instances, and Schema," 2011, <https://arxiv.org/abs/1111.7164>.
- [34] C. Shao, L.-M. Hu, J.-Z. Li et al., "RiMOM-IM: a novel iterative framework for instance matching," *Journal of Computer Science and Technology*, vol. 31, no. 1, pp. 185–197, 2016.
- [35] E. Jiménez-Ruiz and B. C. Grau, "Logmap: logic-based and scalable ontology matching," in *Proceedings of the International Semantic Web Conference*, pp. 273–288, Springer, Bonn, Germany, October 2011.
- [36] N. Pang, W. Zeng, J. Tang, Z. Tan, and X. Zhao, "Iterative Entity Alignment with Improved Neural Attribute Embedding," 2019, https://ceur-ws.org/Vol-2377/paper_5.pdf.
- [37] K. Xu, L. Wang, M. Yu et al., "Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network," 2019, <https://arxiv.org/abs/1905.11605>.
- [38] Y. Cao, Z. Liu, C. Li, J. Li, and T.-S. Chua, "Multi-channel Graph Neural Network for Entity Alignment," 2019, <https://arxiv.org/abs/1908.09898>.

- [39] Q. Zhu, X. Zhou, J. Wu, J. Tan, and L. Guo, *Neighborhood-aware Attentional Representation for Multilingual Knowledge Graphs*, IJCAI, California, CA, USA, 2019.
- [40] Y. Wu, X. Liu, Y. Feng, Z. Wang, and D. Zhao, "Neighborhood Matching Network for Entity Alignment," 2020, <https://arxiv.org/pdf/2011.10132.pdf>.
- [41] S. Pei, L. Yu, G. Yu, and X. Zhang, "Rea: robust cross-lingual entity alignment between knowledge graphs," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2175–2184, California, CA, USA, July 2020.
- [42] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.