

Article

RiQ-KGC: Relation Instantiation Enhanced Quaternionic Attention for Complex-Relation Knowledge Graph Completion

Yunpeng Wang , Bo Ning , Shuo Jiang, Xin Zhou, Guanyu Li and Qian Ma

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; wyp1120220314@dlmu.edu.cn (Y.W.); dotazhanchui@163.com (S.J.); zhouxin314159@163.com (X.Z.); liguanyu@dlmu.edu.cn (G.L.); maqian@dlmu.edu.cn (Q.M.)

* Correspondence: ningbo@dlmu.edu.cn

Abstract: A knowledge graph is a structured semantic network designed to describe physical entities and relations in the world. A comprehensive and accurate knowledge graph is essential for tasks such as knowledge inference and recommendation systems, making link prediction a popular problem for knowledge graph completion. However, existing approaches struggle to model complex relations among entities, which severely hampers their ability to complete knowledge graphs effectively. To address this challenge, we propose a novel hierarchical multi-head attention network embedding framework, called RiQ-KGC, which integrates different-grained contextual information of knowledge graph triples and models quaternion rotation relations between entities. Furthermore, we propose a relation instantiation method for alleviating the difficulty of expressing complex relations between entities. To enhance the expressiveness of relation representation, the relation is integrated by Transformer to obtain multi-hop neighbor information, so that one relation can be embedded into different embeddings according to different entities. Experimental results on four datasets demonstrate that RiQ-KGC exhibits strong competitiveness compared to state-of-the-art models in link prediction, while the ablation experiments reveal that the proposed relation instantiation method achieves great performance.

Keywords: knowledge graph embedding; link prediction; quaternion; complex relation; multi-hop neighbor



Citation: Wang, Y.; Ning, B.; Jiang, S.; Zhou, X.; Li, G.; Ma, Q. RiQ-KGC: Relation Instantiation Enhanced Quaternionic Attention for Complex-Relation Knowledge Graph Completion. *Appl. Sci.* **2024**, *14*, 3221. <https://doi.org/10.3390/app14083221>

Academic Editor: Gianluigi Ferrari

Received: 10 March 2024

Revised: 8 April 2024

Accepted: 9 April 2024

Published: 11 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Knowledge graphs (KGs) represent a structured semantic network used to describe diverse entities and their associations in the physical world, thereby organizing knowledge in a graph structure. In the domains of natural language understanding [1] and knowledge inference [2,3], a comprehensive KG offers significant performance improvements as prior knowledge for downstream tasks such as question answering [4] and recommendation systems [5,6]. However, real-world KGs often represent only a subset of the complete KG, containing a vast amount of undiscovered and poorly organized potential knowledge that is valuable for deep mining and analysis, including node classification [7], node clustering [8], graph query [9–11], entity recognition [12], and link prediction [13]. Consequently, efficient KG representation has become a critical challenge.

A KG consists of a set of entities and a set of relations, with an objective fact represented as a triple comprising entities and relations. Although widely used in large KGs such as WordNet [14], Wikidata [15], and YAGO [16], each is considered to be incomplete, despite containing hundreds of thousands of objective facts.

To address the challenge of effectively representing KGs, certain models [17,18] utilize low-dimensional vectors to represent entities and relations. Through this method, triples are perceived as having potential semantics within the vector space, resulting in entities being distributed across various complex spaces. Relations, on the other hand, are viewed

as transformation patterns between entities such as translation and rotation. These KG representation models can be classified as machine-learning-based, including models such as TransE [19] and RotatE [20]. By embedding entities in more intricate spaces and conveying relational patterns in a more logical manner, these models can achieve a more precise fit to the physical world.

The emergence of deep learning techniques has paved the way for the adoption of large-scale neural networks in extracting entity and relation features. KG representation models that utilize these techniques can be classified as deep-learning-based models such as MTL-KGC [21] and HittER [22]. These models are capable of learning KG features in an automated way, without pre-defining entity and relation representations, resulting in a more flexible modeling approach. However, deep-learning-based models come with certain requirements, such as larger datasets and an increased number of training iterations, that allow the models to fully learn the characteristics of the KG representations.

In this paper, we firstly propose a novel RiQ-KGC framework, which is a KG-embedding model that embeds KGs into hierarchical multi-head attention networks in quaternion space. RiQ-KGC models and extracts geometric information between entities and relations and leverages the attention parameters of the model to generalize the geometric transformation capabilities of entities. The quaternion representation is an ideal choice for smooth rotations and spatial transformations in parametric vector spaces [23], and can accurately represent the symmetric, anti-symmetric, and inverse relations present in the KGs. As a result, RiQ-KGC incorporates the advantages of machine-learning-based models and allows entities to be related to each other using a lower embedding dimension. At the same time, RiQ-KGC offers better learning capability than current deep-learning-based models by leveraging both entity geometry transformation and multi-hop neighborhood information to support model learning.

In the physical world, the number of entities tends to be much greater than the number of relations. However, relations that are expressed as transformation patterns can be too coarse, which restricts the flexibility of transformations and inhibits the geometric representation quality on one-to-one, one-to-n, m-to-n, and m-to-one relations, which we call complex relations. To address this issue, we propose a method of relation instantiation. This method enables the same relation to have different representations in different triples, thereby incorporating additional information about the multi-hop neighboring nodes in the triple. This information represents the context in which the relation is located and enables the relation to be expressed as a transformation pattern that is specific to that context. Employing the relation instantiation method into RiQ-KGC allows the model to capture more useful information when modeling geometric relations, thereby enhancing the embedding performance.

To summarize, we have contributed in the following ways:

- We present RiQ-KGC, a hierarchical multi-head attention network framework that embeds KGs in quaternion spaces and constructs graph embeddings using geometric information.
- We propose a novel relation instantiation method that utilizes contextual information to transform relations, making them express transformation patterns unique to their corresponding triple context.
- We conduct theoretical and empirical analyses that demonstrate the competitiveness of RiQ-KGC against numerous existing models.

The subsequent sections of this paper are structured as follows: Section 2 delves into the advancements of machine learning models and deep learning models in the task of knowledge graph completion. Section 3 presents a detailed exposition of the model architecture of RiQ-KGC. Section 4 conducts comprehensive experiments on RiQ-KGC to demonstrate its superior performance.

2. Related Work

In the realm of graph embedding for KGs, existing research can be roughly categorized into two major families: the machine-learning-based model family and the deep-learning-based model family. The core issue for these models is how to define a scoring function for triples that can be utilized to determine whether a given triple represents an objective fact, serving as the basis for evaluating the quality of graph embeddings. In this section, we will delve into the technical evolution trajectories of these two families.

2.1. Machine-Learning-Based Model

Machine-learning-based models typically adopt a geometric approach, embedding entities into a specific space and using relation vectors to represent the transformation between entities, including translation and rotation. TransE [19] is the most classical model that represents relations as translation patterns. However, TransE has limitations in that translations can only express inverse and symmetric relations, which causes some triples to not be correctly represented. TransH [24], TransR [25], and TransD [26] are all improvements built upon the score function of TransE.

Following translation models, bilinear models such as DisMult [27] and ComplEx [28] were developed, which employ similarity-based scoring functions to gauge the likelihood of triples by matching potential entity and relation semantics in the embedding vector space. DisMult builds on the framework of RESCAL [29], modeling entities as vectors and relations as matrices, with the relation matrix being defined as a diagonal matrix. HolE [30] combines the advantages of RESCAL and DisMult by representing both entities and relations as vectors in space and introducing a loop mechanism for interactions between head and tail entities. This results in a powerful but computationally efficient model. ComplEx, on the other hand, introduces complex embeddings, which represent the embeddings of entities and relations in complex space instead of real space. Additionally, ComplEx modifies the scoring function of the model to an asymmetric form, thereby better capturing asymmetric relations.

Drawing inspiration from complex space bilinear models such as DisMult and ComplEx, RotatE treats relations as rotation operations between entities. This enables relations to express a wide range of attributes, including symmetric, anti-symmetric, inverse, and synthetic attributes. Additionally, TorusE [31], which appeared around the same time as RotatE, defines embeddings as translations on tight Lie groups, embedding entities and relations in spatial torus, thereby eliminating the need for regularization steps. QuatE [23] further developed the concept of complex space and extended it into a quaternion space with two rotation surfaces, allowing for more flexible representation compared to that achievable in a complex space. Additionally, QuatE utilizes Hamilton products to capture internal dependencies among entities and relations, resulting in a simplified and more efficient process for relation modeling. Continuing in this vein, DualE [32] further extended the space to an octonion space, which offers a means for representing both translation and rotation patterns. In recent years, researchers have developed models that embed KGs into more complex spaces. MuRP [33] and MuRE [33] incorporate multi-relation graphs into Poincaré balls in hyperbolic space and transform entity embeddings through the learning of relation-specific parameters via Mobius matrix vector multiplication and Mobius addition. RoTH [34] combines hyperbolic reflections and rotations with attention to simultaneously capture hierarchical and logical patterns in a hyperbolic space. ReTIN [35] introduces real-time influence information on historical facts in hyperbolic geometric space to capture potential hierarchical structures and other rich semantic patterns.

Although some of machine-learning-based models note the difficulty of embedding complex relations, they address the problem posed by single rotation patterns, which cannot adequately model hierarchical structures and relations between entities.

2.2. Deep-Learning-Based Model

Deep-learning-based models provide a flexible approach to extract hidden features of triples using popular frameworks such as convolutional neural networks (CNNs) or Graph Neural Networks (GNNs). A notable example is ConvE [36], which uses relation-specific filters to perform adaptive convolution on the entity transformation matrix, followed by scoring through a dot product of the output and tail embedding. HyperER, proposed by [37], is a hypernetwork architecture that generates simplified specific relation convolutional filters. This approach combines tensor factorization techniques to enhance the performance of the model compared to traditional convolutional models like ConvE. GLSP [38] combines a GNN with an LSTM to effectively store temporal information. By leveraging both long- and short-term networks, GLSP can better capture the complex dynamics of temporal KGs and enhance the predictive ability. Recently, there has been an increasing trend of using attention mechanisms and Transformer [39] frameworks to extract potential connections between entities and relations. For instance, HittER [22] leverages neighbor information extensively and designs a masking and prediction mechanism for source entities to appropriately extract data from the structured Transformer module.

The recent advancement in large-scale pre-trained models in the natural language processing field has inspired researchers to introduce text pre-training models such as BERT [40] to graph embedding. This approach leverages the text descriptions of entities to represent their embeddings, which enhances their semantic information and expressive ability. For example, MTL-KGC [21] introduces an effective multi-task learning method aimed at overcoming the limitations of BERT in the KG domain. By combining relation prediction and correlation sorting tasks with target link prediction, the method achieves stronger performance. StAR [41], on the other hand, proposes a structure-enhanced text representation learning framework for effective KG completion. PKGC [42] discusses the issue of the improper use of pre-trained language models in KG completion, proposing to transform each triplet and its supporting information into natural prompt sentences and further input them into pre-trained language models for classification. More recently, SimKGC [43] adopted a dual-encoder architecture utilizing two parallel BERT modules to model head-entity-relation combinations and tail entities. The model leverages three negative sampling mechanisms and introduces the InfoNCE loss function to make use of comparative learning. KGLM [44] generates a corpus by converting triples in the KG into text sequences, continuing to pre-train language models that have already been pre-trained on other large natural language corpora, and enhancing the model by adding additional entity/relation-type embedding layers to better understand the underlying graph structure.

In fact, deep-learning-based models lack prior rules of feature construction, resulting in lower efficiency in utilizing triples compared to machine-learning-based models. As a result, deep-learning-based models require significant amounts of knowledge triples to learn effective feature construction methods.

The RiQ-KGC proposed in this paper is a deep-learning-based model that integrates geometric information into a new hierarchical Transformer network architecture. It embeds entities into quaternion space and generalizes the rotation between entities using attention mechanism and innovative relation instantiation methods. With regard to embedding complex relations, RiQ-KGC can alleviate the associated difficulties better than machine-learning-based models. Additionally, compared to deep-learning-based models, RiQ-KGC demonstrates more efficient learning capabilities. We evaluated the performance by testing its ability to detect missing entities in triples using link prediction as a downstream task. Our empirical analysis shows that the model is significantly more competitive compared to previous state-of-the-art models.

3. Methodology

In this section, we provide a detailed description of the RiQ-KGC model, which is shown in Figure 1. RiQ-KGC is composed of three functional modules, namely relation instantiation, quaternion space rotation, and geometric decoding. The relation instantiation

module is utilized to localize the relations and make them entity-specific representations. The quaternion space rotation module rotates the entity in quaternion space through the instantiated relation to obtain geometric information. The geometric decoding module analyzes the information obtained from the above two modules. Its output compares the similarity with each entity in the quaternion space, and the entity with the highest score is considered the target result for the triple. Starting with the problem definition of link prediction, we will describe the working mechanism of each module of RiQ-KGC. For ease of comprehension, the primary notations relevant to this paper are enumerated in Table 1.

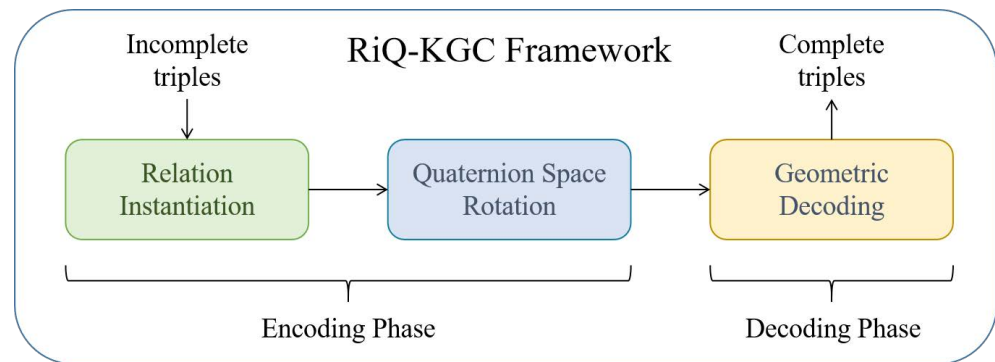


Figure 1. General architecture of RiQ-KGC, which includes three modules: Relation Instantiation, Quaternion Space Rotation, and Geometric Decoding.

Table 1. Notations used in this paper.

Symbol	Description
h, r, t	Head entity, relation, and tail entity in a KG triple
e_h, e_r, e_t	Embeddings of h, r , and t , respectively
e_s	Embeddings of the source entity in a given incomplete triple
e'_s	Embeddings of the predicted entity obtained by quaternion transform
$e_{sec}, e_{thr}, e_{for}$	Embeddings of the second-hop, third-hop, and fourth-hop neighbors
e_{fs}, e_{fr}	Embeddings of the first-hop neighbor's entity and relation
$CLS_0, CLS_1, CLS_f, CLS_{out}$	Different flag vectors
$CLS'_0, CLS'_1, CLS'_f, CLS'_{out}$	Output of CLS_0, CLS_1, CLS_f , and CLS_{out}
T_r	A multi-head Transformer for relation instantiation
T_n	A multi-head Transformer for quaternion space rotation
T_m	A multi-head Transformer for geometric decoding

3.1. Quaternion for Link Prediction

The KG consists of a set of entities E and a set of relations R , which are stored in the database as a series of triples $(h, r, t) \subseteq E \times R \times E$, where $h \in E$ and $t \in E$ are the head and tail entities in the triples, respectively, and $r \in R$ are the relations in the triples. Each triple represents a fact in the KG, indicating that the head entity is connected to the tail entity by the relation, which can be denoted as $h \xrightarrow{r} t$. To better express the semantic meaning of entities and relations, triples in the graph embedding problem are represented as a combination of three d -dimensional vectors (e_h, e_r, e_t) , where $e_h \in E^d$ and $e_t \in E^d$ denote the embedding of the head and tail entities, respectively, and $e_r \in R^d$ denotes the embedding of the relations. The link prediction task aims to discover the missing links between entities in the KG and is therefore considered to play an important role in KG completion.

Each triple is assigned a score, which represents the likelihood that the triple is a true fact. Higher scores indicate that the triple is closer to the true fact. The essence of link prediction is to construct a scoring function that accurately expresses the degree of truth. Given an incomplete triple $(e_h, e_r, ?)$ or $(?, e_r, e_t)$, we need to find the correct head or tail entity. In this paper, we calculate the score as

$$e_t = \arg \max_{i \in E} \phi(e_s, e_r, e_i) \quad (1)$$

where the known entity in the triple is referred to as the source entity e_s and the entity to be predicted is called the target entity e_t . Following Lacroix [45], we unify the prediction tasks as (e_s, e_r, e_t) , where e_s can be either the head entity or the tail entity in the triple, and two different sets of e_r are used to distinguish between the head entity prediction task and the tail entity prediction task. We initialize n_e vectors to represent entities and $2n_r$ vectors to represent relations for the datasets containing n_e entities and n_r relations.

In RiQ-KGC, entities are distributed in quaternion space, and relations are represented as transformation patterns of the entities. Quaternion is a hyper-complex system introduced by Hamilton (Holland, OH, USA). A quaternion Q consists of a scalar q_0 and a set of vectors $q = (q_1, q_2, q_3)$, which can be represented as $Q = q_0 + q = q_0 + q_1i + q_2j + q_3k$, where i, j , and k denote the unit vectors on the x -, y -, and z -axis, respectively. They satisfy $i^2 = j^2 = k^2 = i \cdot j \cdot k = -1$, so that a quaternion can be represented as a quaternion (q_0, q_1, q_2, q_3) . In RiQ-KGC, both the entity embedding $e_s = (e_{s0}, e_{s1}, e_{s2}, e_{s3})$ and the relation embedding $e_r = (e_{r0}, e_{r1}, e_{r2}, e_{r3})$ consist of four parts. The dimension of each part is $d/4$, while the embedding of entities and relations is still represented using d -dimensional vectors.

Quaternions can express ideal parametric smooth rotations that are robust to noise and perturbations. Additionally, a quaternion is able to represent simultaneous rotations on three axes, unlike Eulerian rotation which rotates in a fixed order and can potentially lead to gimbal lock. Zhang [23] demonstrated the advantages of using a quaternion in link prediction, as it possesses similar properties to complex rotations, including the ability to model symmetry, anti-symmetry, and inversion. Furthermore, compared to complex rotation, which only allows a single plane of rotation, a quaternion has two planes and thus provides more degrees of freedom. In the traditional embedding method using quaternion, the relation is represented as a quaternion that describes a rotation pattern. Meanwhile, a source entity can be rotated by the relation using the Hamilton product (quaternion multiplication) as

$$\begin{aligned} e'_s &= e_s \otimes e_r \\ &= (e_{s0}e_{r0} - e_{s1}e_{r1} - e_{s2}e_{r2} - e_{s3}e_{r3}) \\ &\quad + (e_{s0}e_{r1} + e_{s1}e_{r0} + e_{s2}e_{r3} - e_{s3}e_{r2})i \\ &\quad + (e_{s0}e_{r2} - e_{s1}e_{r3} + e_{s2}e_{r0} + e_{s3}e_{r1})j \\ &\quad + (e_{s0}e_{r3} + e_{s1}e_{r2} - e_{s2}e_{r1} + e_{s3}e_{r0})k \end{aligned} \quad (2)$$

This method enables interactions between different parts of the quaternion, leading to richer expressive power.

For link prediction, the similarity between the resulting vector e'_s obtained after rotating e_s and the target entity e_t is commonly used as a scoring function to evaluate the validity of a triple. The similarity between the two entities can be measured by computing the inner product of the corresponding quaternions as

$$\begin{aligned} \phi(e_s, e_r, e'_s) &= e'_s \cdot e_t \\ &= \langle e'_{s0}, e_{t0} \rangle + \langle e'_{s1}, e_{t1} \rangle + \langle e'_{s2}, e_{t2} \rangle + \langle e'_{s3}, e_{t3} \rangle \end{aligned} \quad (3)$$

The traditional method of quaternion embedding imposes a strict constraint on the embedding of entities. Specifically, it expects e'_s to be equal to the e_t in the triple, which maximizes the objective function $\phi(e_s, e_r, e'_s)$. Nevertheless, in reality, each entity might be rotated by numerous other entities through different relations, and the constraint imposed by numerous neighboring entities can lead to a Pareto optimal scenario for the final embedding position of the entity. In such a scenario, the objective is to ensure that each true triple containing a score is as high as possible, rather than aiming to make each of them achieve the highest score. However, this constraint can have a significant impact on the model performance, particularly when the entities are more densely distributed. Figure 2

shows how RiQ-KGC leverages a large number of parameters to establish a mapping relation between e'_s and e_t , enabling the deep learning network to capitalize on its powerful fitting ability, while still retaining the interpretability advantages of the geometric approach. Additionally, we utilize Transformer to incorporate contextual information and enable the model to more accurately capture the entity's information with the increase in knowledge.

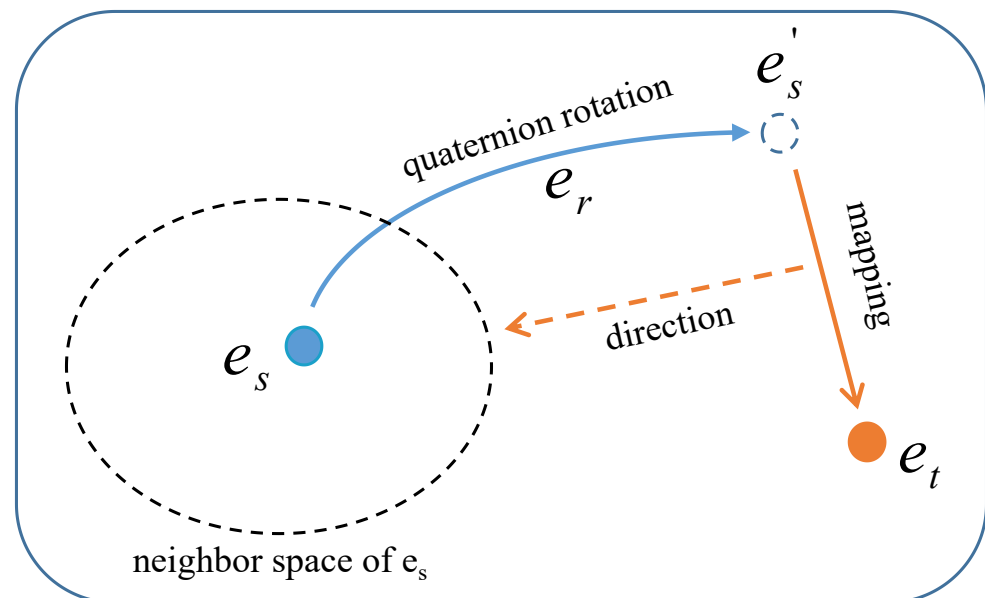


Figure 2. Visualization process of entity transformation.

3.2. Relation Instantiation

The expressiveness of relations is a critical factor in determining the effectiveness of graph-embedding models. As relations represent patterns of transformation between entities, they play a crucial role in the representation of geometric information. In most KGs, the number of entities is significantly larger than the number of relations. For example, the WN18RR [36] dataset contains 40,943 instances of entities with 93,003 triples, while only 11 relations are given. As a result, each entity has only 11 geometric transformations, even though the number of neighbors can be much higher. Instantiating relations to corresponding entities can alleviate the m-to-1 problem in entities and the problem of complex relations between them. In the following section, we will demonstrate the effectiveness of the relation instantiation method in two different cases.

Case 1. Suppose that Hank, Ross, and Bruce are John's father, mother, and uncle, respectively, and that all three of them are doctors at the Johns Hopkins Hospital (JHP). We can represent the relations between them using a relational rotation graph, as shown in Figure 3a. The relations for father, mother, and uncle are different, and therefore their corresponding embedding positions are also different. However, the *Work_for* relation forces JHP to be in three potential embedding positions simultaneously, which is not possible.

Figure 3b shows that the *Work_for* relation can represent different rotations after the relation instantiation, and thus the embedding position of JHP can be accurately represented. This illustrates how relation instantiation can alleviate the m-to-1 problem. The process of relation instantiation can be seen as a form of "reverse clustering", which prevents similar entities from being embedded in close proximity due to m-to-1 relations. This approach enhances the model's ability to distinguish and accurately classify the hardest negative samples.

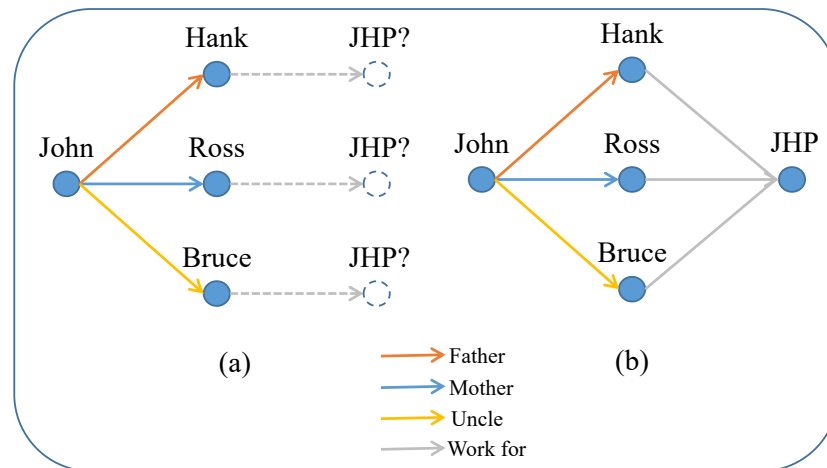


Figure 3. M-to-1 relation representation of entities. Subfigure (a) shows the inaccurate expression of a single relation pattern in M-to-1 relations, while subfigure (b) improves this situation through relation instantiation method. The arrows in different colors indicate the different rotation between entities, with solid lines representing the determined rotation process and dashed lines representing the indeterminate rotation process.

Case 2. As shown in Figure 4a, when John grows up and becomes a doctor at JHP, Hank is not only his father but also his colleague. However, since father and colleague are represented by two different relations that correspond to different rotations, an error can occur if Hank is assigned two possible embedding positions simultaneously. One potential approach to address this issue involves making two relations represent the same rotation, which can uniquely position Hank, albeit in a way that is not equivalent in the physical world.

Figure 4b illustrates how two relations can be used to express the rotation from John to Hank using different rotations, without affecting other entities when the relations are applied after the relation instantiation. This approach enables us to uniquely position Hank and resolve the issue of having two possible embedding positions simultaneously.

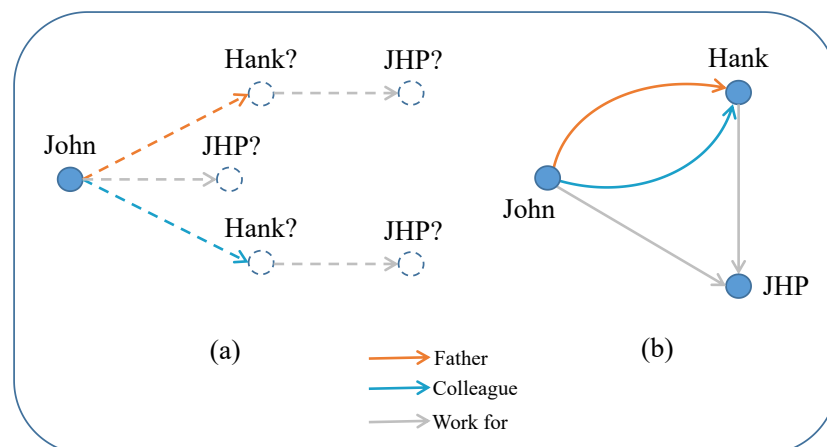


Figure 4. Representation of complex relations between entities. Subfigure (a) shows the inaccurate expression of complex relations, while subfigure (b) improves this situation through relation instantiation method. The arrows in different colors indicate the different rotation between entities, with solid lines representing the determined rotation process and dashed lines representing the indeterminate rotation process.

The contextual information of an entity can be reflected by considering its neighboring entities in the KG, also known as “neighbors”. We designed a hierarchical Transformer structure to enable relations to fully integrate the environmental information contained

within a given entity's neighbors. Our goal was to ensure that relations can have different rotations in different environmental contexts. Figure 5 shows the specific construction of the relation instantiation module. In relation instantiation, we select a significant number of second-hop, third-hop, and fourth-hop neighbors, represented by e_{sec} , e_{thr} , and e_{for} , to provide contextual information for the source entity. The n th-hop neighbors refer to the nodes that the source entity can reach through n transformations of relations. In this context, each transformation of the source entity to the tail entity through a relation in the triple is considered a first-hop. Neighbors with smaller hop counts, such as 1-hop and 2-hop neighbors, have a closer relationship with the source entity. They can directly reflect relevant information about the source entity. On the other hand, neighbors with larger hop counts, such as 3-hop and 4-hop neighbors, can provide additional information by roughly reflecting the scene where the source entity is located. It is important to note that we do not use the relations of these neighboring triples, as the size of the relation set R is significantly smaller than the number of neighboring triples we will select. As a result, these relations do not contribute to the feature representation of the source entity.

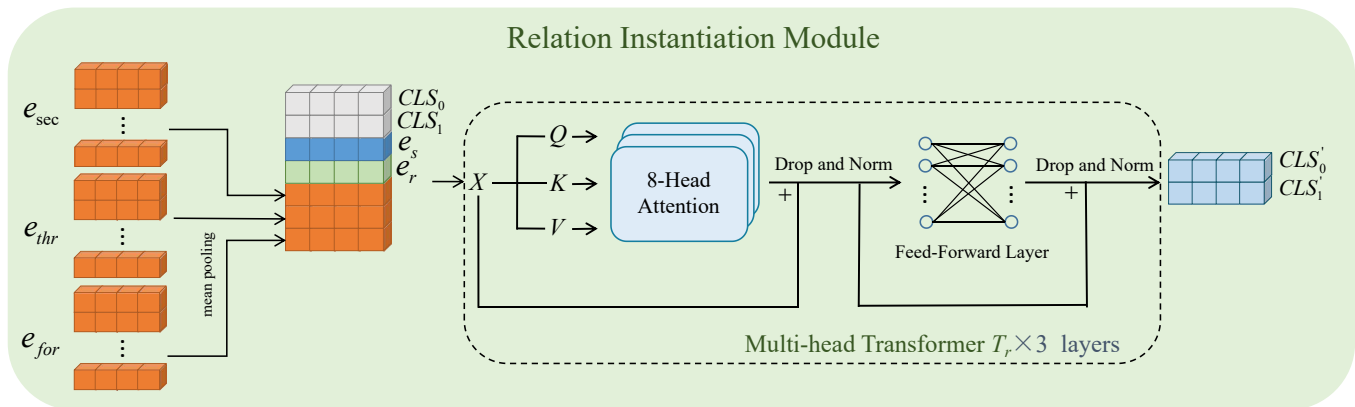


Figure 5. Architecture of relation instantiation module.

Then, we obtain three $1 \times d$ context vectors by averaging the representations of e_{sec} , e_{thr} , and e_{for} . These context vectors are then inputted into the multi-headed Transformer T_r along with the source entity e_s , the relation e_r , and the two flag vectors, CLS_0 and CLS_1 . The resulting two relation components that are obtained via T_r are represented by CLS'_0 and CLS'_1 , which represent the instantiation results of the relation from different perspectives. Using multiple relation components is intended to convey more contextual information and to help balance the weight of information across multi-hop neighbors during decoding.

The multi-head Transformer consists of a sub-layer that comprises a multi-head attention network connected in series with a sub-layer of a feed-forward network. The two sub-layers are connected by a connection layer that includes residual connections, dropout, and normalization operations.

3.3. Quaternion Space Rotation

Figure 6 shows the specific construction of the Quaternion space rotation module. We combined the outputs CLS'_0 and CLS'_1 with e_r to obtain a relation instantiation matrix. We performed a Hamilton product of e_s with each of the three vectors in this matrix, resulting in a quaternion rotation of the source entity from three different angles and producing a quaternion matrix M_q as

$$M_q = \text{concat}(e_s \otimes e_r, e_s \otimes CLS'_0, e_s \otimes CLS'_1) \quad (4)$$

M_q represents multi-hop neighbor information of the source entity, capturing a coarse-grained contextual situation. Meanwhile, fine-grained neighbor information is derived from the first-hop neighbors of the source entity. To integrate and unify these different levels

of contextual information, we link the first-hop neighbor information with M_q . The first-hop neighbors serve as intermediate information that are used to explicitly establish the association between the source entity and its context, thus creating hierarchical contextual information. Inspired by Chen [22], we utilize a multi-head Transformer T_n to integrate the first-hop neighbor entities e_{fs} and the relations e_{fr} . The source entity and relation, e_s and e_r , respectively, are also used as input to ensure that the information of the triple can be fully processed. Thus, each input group consists of the flag vector CLS_n , together with e_s or e_{fs} and e_r or e_{fr} , which are sequentially fed into T_n . The first-hop neighbor information is combined through T_n using CLS_f , and their corresponding outputs CLS'_f are concatenated and stored as the neighbor matrix M_n as

$$M_n = \text{concat}\left(\sum_{i=1}^{f+1} CLS'_{fi}\right) \quad (5)$$

where f is the number of first-hop neighbors. To ensure that the source entity information is not overwhelmed by excessive neighbor information, e_s in T_n is replaced or masked by a random entity with a certain probability, encouraging the model to recover the source entity later.

M_n conveys the source entity information in a fine-grained environment, while M_q conveys the target entity information produced by rotation in a coarse-grained environment. We combine M_n with M_q to produce a mixed matrix M_m as

$$M_m = \text{concat}(CLS_{out}, M_n, M_q) \quad (6)$$

Consequently, it covers the complete process of quaternion rotation under the influence of contextual relation. We insert a flag vector CLS_{out} onto the first line of M_m , which is subsequently employed in geometric decoding to resolve the representation of the target entity.

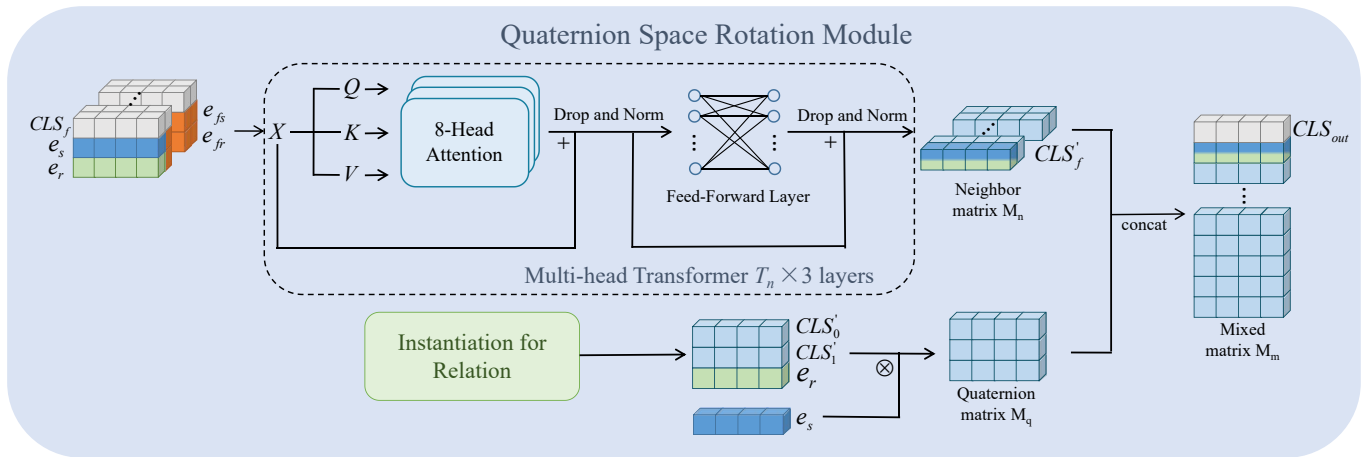


Figure 6. Architecture of quaternion space rotation module.

3.4. Geometric Decoding

As shown in Figure 7, M_m is processed by a multi-head Transformer T_m for decoding. The first two lines of the resulting outputs, CLS'_{out} and e_{pred} , are applied for resolving the representation of the target entity and the degree of entity reduction, respectively. To provide the model with additional fitting space, CLS'_{out} is finally inputted into a linear feed-forward network as

$$e_o = W \times CLS'_{out} + b \quad (7)$$

where W and b correspond to the weight and bias, and the output e_o represents the target entity. Subsequently, we compute the similarity with each $i \in E$ through dot product as

$$\phi(e_s, e_r, e_i) = e_o \cdot e_i \quad (8)$$

The similarity serves as the score of the triple, which can be perceived as a confidence level when the tail entity of the triple corresponds to e_i . The higher the score of the triple, the more likely it is that the entity is the target entity and is therefore appropriately predicted.

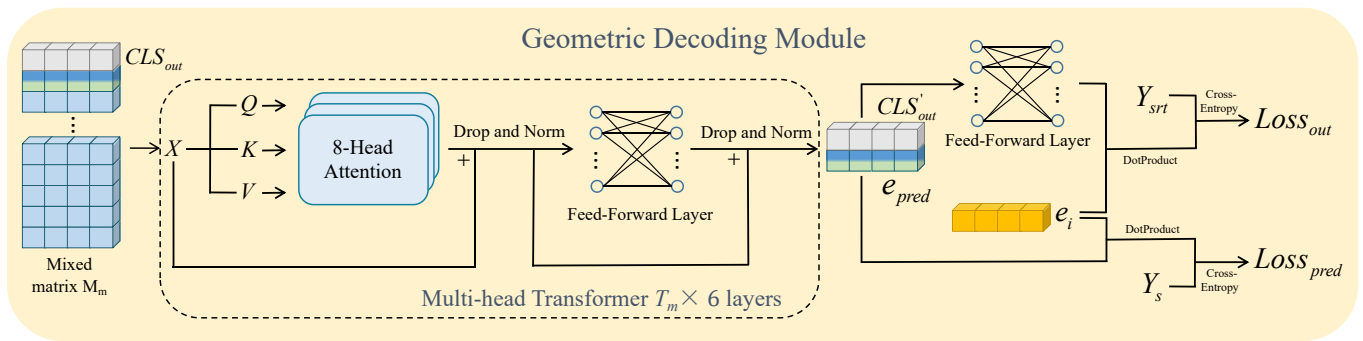


Figure 7. Architecture of geometric decoding module.

During the training process, the score is established as the probability distribution of the correct entity based on the softmax activation function. This probability distribution is implemented to calculate the cross-entropy loss as

$$Loss_{out} = - \sum_{i=1}^E Y_{srt} \cdot \text{softmax}(\phi(e_s, e_r, e_i)) \quad (9)$$

where $Y_{srt} \in \{0, 1\}$ and Y_{srt} is 1 if i is the true target entity.

To avoid overemphasizing neighbor information and disregarding source entity information during decoding, we use e_{pred} to reduce the impact of the source entity. This is achieved by calculating similarity through the dot product for each $i \in E$. The degree of reduction in e_{pred} to the source entities is measured using cross-entropy loss as

$$Loss_{pred} = - \sum_{i=1}^E Y_s \cdot \text{softmax}(e_{pred}, e_i) \quad (10)$$

where $Y_s \in \{0, 1\}$ and Y_s is 1 if i is the source entity.

We obtain the final loss value by adding the two loss values with weights as

$$Loss = \alpha Loss_{out} + \beta Loss_{pred} + \gamma \|E\|_2 \quad (11)$$

Additionally, we apply entity regularization to deter over-fitting and to generalize the embedding location of the entity.

4. Experiments and Numerical Results

This section presents detailed comparative experiments, exploring the competitive ability of RiQ-KGC for link prediction. We provide an enumeration of the application and experimental configuration of the datasets. We also conducted ablation experiments to evaluate the ability of each model component to contribute to performance. Furthermore, we investigate the model's capacity to capture complex relations.

4.1. Experimental Setup

4.1.1. Datasets

We utilized a diverse set of four datasets, as outlined in Table 2. These datasets were specifically chosen due to their distinct scales, enabling a comprehensive demonstration of the model's capabilities across various scenarios.

FB15K-237 [46] is a subset of the widely used Freebase [47]. Its entities consist of those mentioned more frequently than 100 times in FreeBase entities. To avoid information leakage caused by test data in the training set, FB15K-237 filters the datasets. Similarly, WN18RR is a subset of WordNet that avoids the same issue by selecting entities exclusively from WordNet. Evaluating models on both FB15K-237 and WN18RR could be beneficial for comparing the performance comprehensively, which are widely used in link prediction tasks.

Table 2. Dataset settings.

Datasets	E	R	Training	Validation	Test
FB15K-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3034	3134
CoDEX-M	17,050	51	185,584	10,310	10,311
CoDEX-L	77,951	69	551,193	30,622	30,622

The CoDEX-M and CoDEX-L datasets are subsets derived from CoDEX [48], a collection of KG completion datasets obtained from Wikidata and Wikipedia. CoDEX provides a broader and more diverse range of content, resulting in richer relations and a more balanced distribution of relation types. Furthermore, there are distinct characteristics that differentiate CoDEX-M from CoDEX-L in terms of relation types. CoDEX-M consists of a higher proportion of symmetric relations, while CoDEX-L contains a larger number of composition relations. Evaluating models on both the CoDEX-M and CoDEX-L datasets offers valuable insights into their performance in challenging and complex scenarios.

4.1.2. Evaluation Metrics

In link prediction, the aim is to rank all entities in the triple according to their probability of being the correct target. The model's performance is evaluated based on the ranking position of the actual target entity. To assess the model's overall recall, we employed Mean Reciprocal Rank (MRR), which measures the average proportion of correctly ranked entities across all test cases. We report the MRR and ranking results in a filtered setting, where entities other than the ground-truth target are filtered out.

4.1.3. Baselines

For the FB15K237 and WN18RR datasets, we carefully selected models from different families between 2015 and 2023, whose results were taken from their original paper, to serve as baselines. In the machine-learning-based model family, we chose classical models such as DisMult, RotatE, QuatE, MuRE, and MuRP. These models employ spatial geometric embeddings and have been widely utilized in KG completion tasks. In the deep-learning-based model family, we selected the classical convolutional neural network model ConvE, as well as recent models including HypER, MTL-KGC, StAR, and KGLM. All of these models utilize Transformer architectures, which have achieved state-of-the-art performance and have significantly contributed to the advancement of KG link prediction.

The aforementioned models have not been experimentally recorded on the CoDEX-M and CoDEX-L datasets. Therefore, we selected models and experimental results from the original CoDEX article, namely TransE, RESCAL, ComplEx, ConvE, and TuckER. This choice was made to ensure consistency with previous research and to maintain comparability by using the same benchmarks for evaluation.

4.1.4. Implementation Details

Our code was implemented using the PyTorch (Version 1.13.1, Facebook AI Research, Menlo Park, CA, USA) [49] framework and was built upon the LibKGC [50] library. We utilized a multi-head Transformer architecture in both T_n and T_r , which consists of three multi-head Transformers, and T_m , which contains six multi-head Transformers. Each multi-head Transformer consists of eight heads with a feed-forward network dimension of 1280, and the hidden layer dropout rate is set to 0.1.

To train the datasets, we initialized entity- and relation-embedding dimensions to 320 and established a batch size of 256 samples. Table 3 displays the detailed training parameter configurations for the four datasets, while the determination of the parameters relating to the number of multi-hop neighbors was based on the guidelines presented in Section 4.4. We employed a grid search strategy to obtain the optimal configuration of hyperparameters, where α , β , and γ were searched within the set $\{0.1, 0.5, 0.8, 1\}$. Finally, the optimal parameters for α , β , and γ were determined to be 1, 0.8, and 0.1, respectively. We applied the Adamax optimization strategy [51] with a learning rate set to 0.01. The rate started at 0, linearly increased during the first 10% of epochs, and linearly decreased in subsequent epochs. Our experiments ran on RTX 4060 8G graphics cards using the above configuration.

Table 3. Settings of the number of multi-hop neighbors.

Datasets	1-Hop	1-Hop Dropout	2-Hop	3-Hop	4-Hop
FB15K-237	40	0.3	50	150	500
WN18RR	30	0.5	30	100	250
CoDEx-M	40	0.3	50	150	500
CoDEx-L	20	0.5	30	100	250

After training RiQ-KGC, we performed additional fine-tuning steps on the model. Firstly, we fixed the entity- and relation-embedding representation and fine-tuned the model with a learning rate of 0.003. Subsequently, we fixed all Transformer module parameters and fine-tuned only the final linear layer using a learning rate of 0.001.

4.2. Comparison with Existing Methods

Table 4 presents a comparison between RiQ-KGC and the baseline models for the FB15K-237 and WN18RR datasets. The results show that RiQ-KGC demonstrates notable advantages in terms of the MRR, H@1, and H@3 indicators for both datasets. Notably, RiQ-KGC exhibited a remarkable performance improvement of 1.5% in the MRR indicator for the WN18RR dataset. This highlights RiQ-KGC's superior precision in identifying target entities. However, it should be noted that RiQ-KGC under-performed compared to StAR and KGLM in the H@10 indicator. While the BERT-based models excel in extracting extensive entity feature information from textual descriptions, thereby enhancing rough recall rates, the presence of noise stemming from these descriptions poses a challenge to achieving precise recall. The relation instantiation and quaternion transform mechanism employed in RiQ-KGC contribute to a sparser distribution between entities, facilitating the precise identification of relevant entities. This characteristic grants RiQ-KGC a notable advantage over non-BERT-based models in H@10.

Table 5 presents a performance comparison between RiQ-KGC and the baseline model on CoDEx-M and CoDEx-L. As shown in Table 5, RiQ-KGC outperforms the baseline model in all evaluation metrics, particularly exhibiting a remarkable performance improvement of 0.8% and 1.8% in the H@10 indicator, for the CoDEx-M and CoDEx-L datasets, respectively, where it demonstrates the greatest advantage. This suggests that RiQ-KGC can maintain good recall performance even in challenging scenarios and, as a result, outperform other models.

Table 4. Comparison of model results on FB15K-237 and WN18RR datasets, where the bold font represents the optimal result and the underlined font represents the sub-optimal result. The results of these baselines are taken from their original article.

Model	FB15K-237				WN18RR			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
DisMult [27]	0.241	0.155	0.263	0.419	0.430	0.390	0.440	0.490
RotatE [20]	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571
QuatE [23]	0.311	0.221	0.342	0.495	<u>0.481</u>	0.436	0.500	0.564
MuRP [33]	0.335	0.243	0.367	0.518	<u>0.481</u>	<u>0.440</u>	0.495	0.566
MuRE [33]	0.336	0.245	0.370	0.521	0.475	0.436	0.487	0.554
ConvE [36]	0.325	0.237	0.356	0.501	0.430	0.400	0.440	0.520
HypER [37]	<u>0.341</u>	<u>0.252</u>	<u>0.376</u>	0.520	0.465	0.436	0.477	0.522
MTL-KGC [21]	0.267	0.172	0.298	0.458	0.331	0.203	0.383	0.597
StAR [41]	0.296	0.205	0.322	0.482	0.401	0.243	0.491	<u>0.709</u>
KGLM [44]	0.289	0.200	0.314	0.468	0.467	0.330	0.538	0.741
RiQ-KGC (ours)	0.346	0.257	0.378	<u>0.524</u>	0.496	0.457	<u>0.509</u>	0.572

Table 5. Comparison of model results on the CoDEX-M and CoDEX-L datasets, where the bold font represents the optimal result and the underlined font represents the sub-optimal result. The results of these baselines are taken from the original article of the CoDEX datasets.

Model	FB15K-237				WN18RR			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE [19]	0.303	0.223	0.336	0.454	0.187	0.116	0.218	0.317
RESCAL [29]	0.317	0.244	0.347	0.456	0.304	<u>0.242</u>	0.331	0.419
ComplEx [28]	<u>0.337</u>	0.262	0.370	<u>0.476</u>	0.294	0.237	0.318	0.400
ConvE [36]	0.318	0.239	0.355	0.464	0.303	0.240	0.330	0.420
TuckER [45]	0.328	<u>0.259</u>	0.360	0.458	<u>0.309</u>	0.244	<u>0.339</u>	<u>0.430</u>
RiQ-KGC (ours)	0.339	0.262	<u>0.368</u>	0.484	0.313	<u>0.242</u>	0.340	0.448

4.3. Parameter Evaluation

We hypothesized that the quantity of multi-hop neighbor information may have an impact on relation instantiation. Figure 8 illustrates the impact of the number of one-hop neighbors (left figure) and the number of multi-hop neighbors (right figure) on model performance. For the one-hop neighbors, the number of neighbors significantly influences the model's performance, and the optimal number varies across different datasets. The FB15K237 and CoDEX-M datasets exhibit optimal performance with 40 one-hop neighbors, while the WN18RR and CoDEX-L datasets perform best with 30 and 20 one-hop neighbors, respectively. The experimental results across the four datasets indicate that the increase in the number of one-hop neighbors correlates positively with the model's performance, particularly when the number of neighbors is small. However, beyond a certain performance threshold, the additional increase in one-hop neighbors no longer contributes significantly to the model's performance.

For the multi-hop neighbors, we conducted further experiments with varying multi-hop neighbor settings, as listed in Table 6. For the multi-hop neighbors, their influence on the model's performance was not substantial. This is attributed to the fact that the embeddings of multi-hop neighbors are indirectly utilized in the instantiation for relation modules through an average pooling operation, which is not directly employed in constructing the features of the target entity. In terms of specific datasets, FB15K237 and CoDEX-L achieved their best performance in Setting-2, whereas WN18RR and CoDEX-M attained their optimal performance in Setting-3.

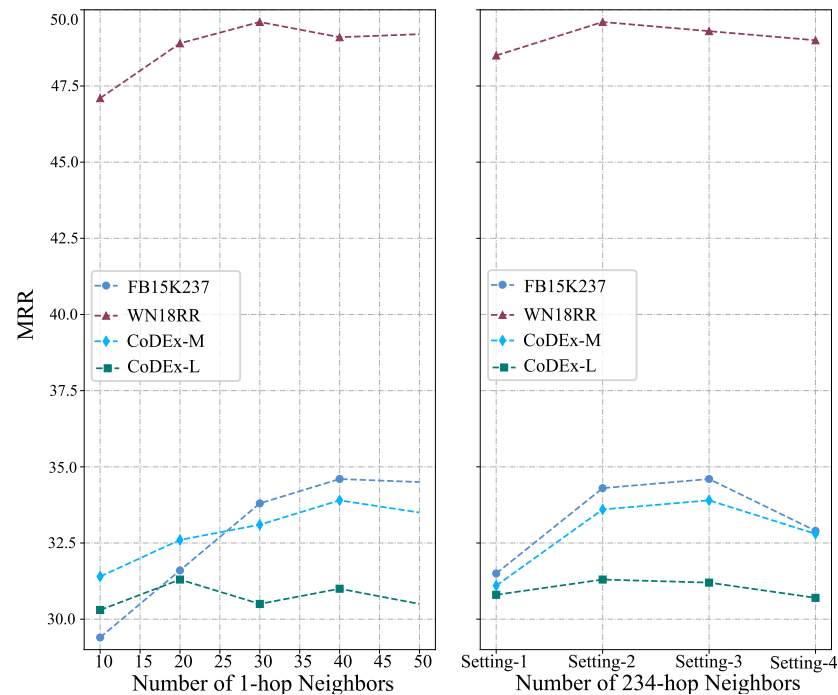


Figure 8. Impact of multi-hop neighbors on four datasets.

Table 6. Settings of multi-hop neighbors.

	Second-Hop	Third-Hop	Fourth-Hop
Setting-1	10	25	60
Setting-2	30	100	250
Setting-3	50	150	500
Setting-4	80	200	750

4.4. Ablation Studies

We conducted ablation experiments on all four datasets to assess the contribution of each module in RiQ-KGC. Apart from the complete model, we designed four additional ablation models. “RiQ-KGC w/o allN” removes all neighboring inputs in each module. “RiQ-KGC-w/o-234N” excludes multi-hop neighbor inputs in the instantiation for the relation module. “RiQ-KGC-w/o-Q” eliminates the quaternion rotation step in the quaternion space rotation module and replaces the corresponding input position with the source entity. “RiQ-KGC-w/o-R” discards the instantiation for the relation module and utilizes original relation to rotate the source entity.

Figure 9 presents the performance of each ablation model on the four datasets using the MRR metric, which can show the overall recall rate of samples. The most significant conclusion is that “RiQ-KGC” achieved the best performance, while “RiQ-KGC w/o all N” exhibited the worst performance, indicating the crucial role of neighboring nodes. The efficacy of “RiQ-KGC-w/o-R” was generally poor, highlighting the importance of relation instantiation methods. “RiQ-KGC-w/o-234N” outperformed “RiQ-KGC-w/o-R” in all datasets, demonstrating that multi-hop neighbor information enhances the relation instantiation method. The impact of “RiQ-KGC-w/o-Q” is highly uncertain because the proportion of complex relations in the four datasets varied and quaternion rotation patterns have a greater advantage when applied to datasets with a higher ratio of one-to-one relations.

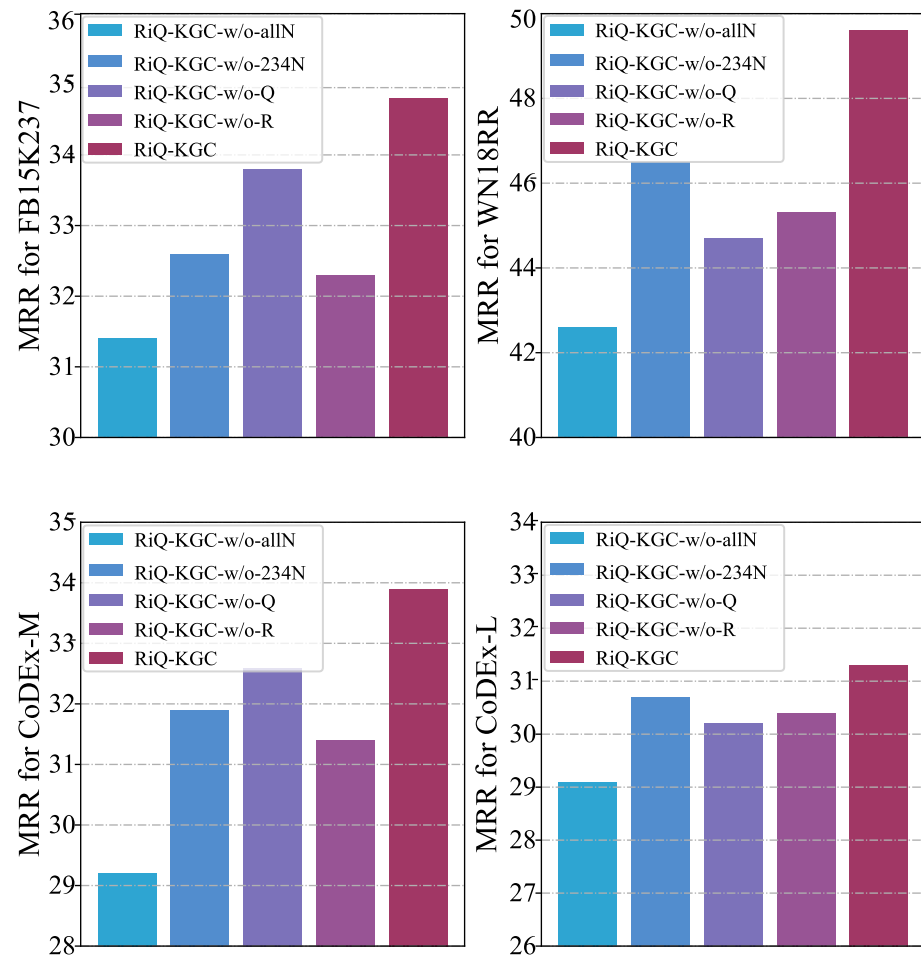


Figure 9. Ablation study on each component.

4.5. Modeling Capability for Complex Relation

To better demonstrate the model's ability to handle complex relations, we conducted visual experiments to separately evaluate the modeling speed and effectiveness of the ablation model for four types of complex relations. Based on Figure 10, the four graphs in each column correspond to the modeling process of the ablation model for four distinct complex relation types on the same datasets. Meanwhile, the four graphs in each row provide a comparison of modeling for the same complex relation type across the four datasets.

From the results, it is evident that there is no significant difference in the modeling speed and effectiveness of each ablation model for the one-to-one and m-to-n relations, indicating that each ablation model does not have a specialized ability to handle these two complex relations. However, there are significant differences in the modeling processes of one-to-n and m-to-one relations among different ablation models. Specifically, "RiQ-KGC" and "RiQ-KGC w/o-Q" usually perform better, whereas "RiQ-KGC w/o-234N" and "RiQ-KGC w/o-R" usually perform worse, and "RiQ-KGC w/o-all-N" is the worst in all scenarios. This demonstrates the targeted contribution ability of "RiQ-KGC-w/o-234N" and "RiQ-KGC-w/o-R" to complex relationships, which is consistent with the theory in Section 3.2.

Additionally, the one-hop neighbor was found to have a strong overall contribution ability to the model, consistent with the results of the ablation experiment.

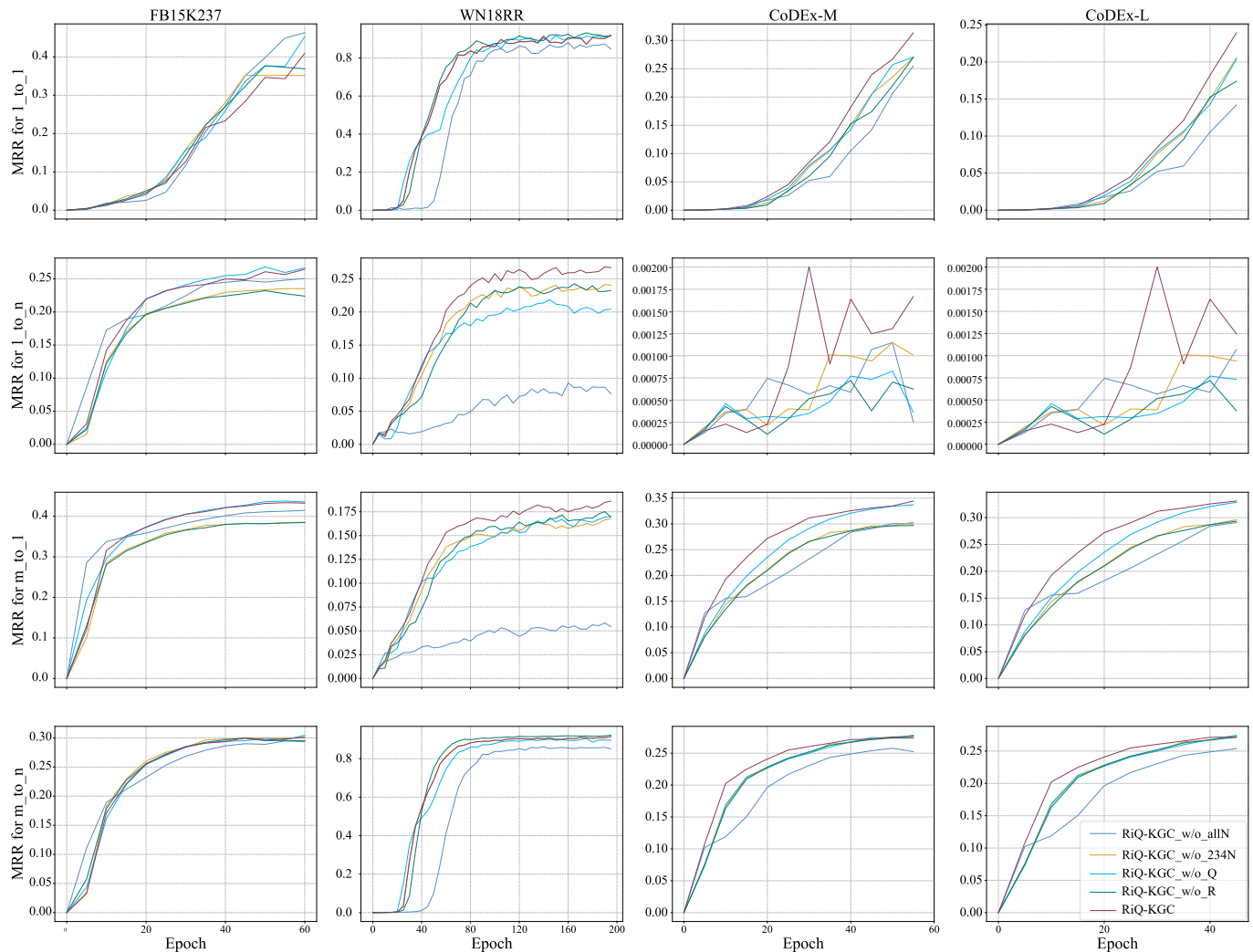


Figure 10. Comparison of modeling capabilities for complex relations on four datasets. All subplots share the same legends. To avoid cluttering the layout, we set the legends in the bottom right corner of the last subplot.

5. Conclusions

The present study proposes a novel knowledge-graph (KG)-embedding framework called RiQ-KGC, which utilizes geometric rotation information from the quaternion space and integrates multi-hop neighbor information into relation representation using a relation instantiation method. The proposed model builds upon the strengths of quaternion space embedding by enabling the representation of symmetric, anti-symmetric, and inverse relations between entities and enhancing the ability to represent complex relations. Moreover, RiQ-KGC learns intrinsic connections in KGs by applying a large number of parameters and strengthening the association between rotation results and target entities via parameter mapping. Consequently, entities and relations can be more accurately expressed.

In the future, we aim to explore more accurate and efficient methods of representing neighbor information to support more powerful relation instantiation methods. Entity transformation via relations is a classical notion in graph embedding and integrating entity transformation with advanced deep learning frameworks is a promising direction for future research. We will continue exploring methods to represent entity transformation in deep learning models with the goal of building more advanced and effective embedding models for link prediction.

Author Contributions: Conceptualization, Y.W. and B.N.; methodology, Y.W.; software, Y.W. and S.J.; validation, S.J.; formal analysis, X.Z.; investigation, Y.W.; resources, X.Z.; data curation, B.N.; writing—original draft preparation, Y.W.; writing—review and editing, B.N.; visualization, S.J.; supervision, G.L.; project administration, B.N.; funding acquisition, G.L. and Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61976032, and the National Natural Science Foundation of China, grant number 62002039.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Hayashi, H.; Hu, Z.; Xiong, C.; Neubig, G. Latent Relation Language Models. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 7911–7918.
- Verga, P.; Sun, H.; Soares, L.B.; Cohen, W.W. Adaptable and Interpretable Neural Memory Over Symbolic Knowledge. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 3678–3691.
- Song, X.; Li, J.; Cai, T.; Yang, S.; Yang, T.; Liu, C. A survey on deep learning based knowledge tracing. *Knowl. Based Syst.* **2022**, *258*, 110036. [\[CrossRef\]](#)
- Bi, X.; Nie, H.; Zhang, G.; Hu, L.; Ma, Y.; Zhao, X.; Yuan, Y.; Wang, G. Boosting question answering over knowledge graph with reward integration and policy evaluation under weak supervision. *Inf. Process. Manag.* **2023**, *60*, 103242. [\[CrossRef\]](#)
- Hu, X.; Xu, J.; Wang, W.; Li, Z.; Liu, A. A graph embedding based model for fine-grained POI recommendation. *Neurocomputing* **2021**, *428*, 376–384. [\[CrossRef\]](#)
- Liu, H.; Tong, Y.; Han, J.; Zhang, P.; Lu, X.; Xiong, H. Incorporating Multi-Source Urban Data for Personalized and Context-Aware Multi-Modal Transportation Recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 723–735. [\[CrossRef\]](#)
- Wang, X.; Cui, P.; Wang, J.; Pei, J.; Zhu, W.; Yang, S. Community Preserving Network Embedding. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31, pp. 203–209.
- Nie, F.; Zhu, W.; Li, X. Unsupervised Large Graph Embedding. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31, pp. 2422–2428.
- Yan, J.; Gu, Z.; Jiang, Z.; Gao, C.; Yang, J. Persistent Graph Stream Summarization For Real-time Graph Analytics. *World Wide Web J.* **2023**, *31*, 7911–7918.
- Yi, P.; Li, J.; Choi, B.; Bhowmick, S.S.; Xu, J. FLAG: Towards Graph Query Autocompletion for Large Graphs. *Data Sci. Eng.* **2022**, *7*, 175–191. [\[CrossRef\]](#)
- Liu, P.; Wang, X.; Fu, Q.; Yang, Y.; Li, Y.F.; Zhang, Q. KGVQL: A knowledge graph visual query language with bidirectional transformations. *Knowl.-Based Syst.* **2022**, *250*, 108870. [\[CrossRef\]](#)
- Shaan, K. A Survey of Arabic Named Entity Recognition and Classification. *Comput. Linguist.* **2014**, *40*, 469–510. [\[CrossRef\]](#)
- Wu, H.; Song, C.; Ge, Y.; Ge, T. Link Prediction on Complex Networks: An Experimental Survey. *Data Sci. Eng.* **2022**, *7*, 253–278. [\[CrossRef\]](#) [\[PubMed\]](#)
- Miller, G.A. WordNet: A Lexical Database for English. *Communications* **1995**, *38*, 39–41. [\[CrossRef\]](#)
- Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Communications* **2014**, *57*, 78–85. [\[CrossRef\]](#)
- Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A core of semantic knowledge. In Proceedings of the 16th International Conference on World Wide Web, Banff, AB, Canada, 8–12 May 2007; pp. 697–706. [\[CrossRef\]](#)
- Chen, Z.; Wang, X.; Wang, C.; Li, J. Explainable Link Prediction in Knowledge Hypergraphs. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 262–271.
- Zhang, Y.; Sheng, M.; Zhou, R.; Wang, Y.; Han, G.; Zhang, H.; Xing, C.; Dong, J. HKGB: An Inclusive, Extensible, Intelligent, Semi-auto-constructed Knowledge Graph Framework for Healthcare with Clinicians' Expertise Incorporated. *Inf. Process. Manag.* **2020**, *57*, 102324. [\[CrossRef\]](#)
- Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
- Sun, Z.; Deng, Z.; Nie, J.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

21. Kim, B.; Hong, T.; Ko, Y.; Seo, J. Multi-Task Learning for Knowledge Graph Completion with Pre-trained Language Models. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 1737–1743. [\[CrossRef\]](#)
22. Chen, S.; Liu, X.; Gao, J.; Jiao, J.; Zhang, R.; Ji, Y. HittER: Hierarchical Transformers for Knowledge Graph Embeddings. In Proceedings of the Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 10395–10407. [\[CrossRef\]](#)
23. Zhang, S.; Tay, Y.; Yao, L.; Liu, Q. Quaternion Knowledge Graph Embeddings. In Proceedings of the Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 2731–2741.
24. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
25. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.
26. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge Graph Embedding via Dynamic Mapping Matrix. In Proceedings of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015; pp. 687–696. [\[CrossRef\]](#)
27. Yang, B.; Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
28. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 2071–2080.
29. Nickel, M.; Tresp, V.; Krieger, H. A Three-Way Model for Collective Learning on Multi-Relational Data. In Proceedings of the International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 809–816.
30. Nickel, M.; Rosasco, L.; Poggio, T.A. Holographic Embeddings of Knowledge Graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1955–1961.
31. Ebisu, T.; Ichise, R. TorusE: Knowledge Graph Embedding on a Lie Group. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1819–1826.
32. Cao, Z.; Xu, Q.; Yang, Z.; Cao, X.; Huang, Q. Dual Quaternion Knowledge Graph Embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; pp. 6894–6902.
33. Balazevic, I.; Allen, C.; Hospedales, T.M. Multi-relational Poincaré Graph Embeddings. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 4465–4475.
34. Chami, I.; Wolf, A.; Juan, D.; Sala, F.; Ravi, S.; Ré, C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In Proceedings of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6901–6914. [\[CrossRef\]](#)
35. Jia, Y.; Lin, M.; Wang, Y.; Li, J.; Chen, K.; Siebert, J.; Zhang, G.Z.; Liao, Q. Extrapolation over temporal knowledge graph via hyperbolic embedding. *CAAI Trans. Intell. Technol.* **2023**, *8*, 418–429. [\[CrossRef\]](#)
36. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2D Knowledge Graph Embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1811–1818.
37. Balazevic, I.; Allen, C.; Hospedales, T.M. Hypernetwork Knowledge Graph Embeddings. In Proceedings of the 28th International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; Volume 11731, pp. 553–565. [\[CrossRef\]](#)
38. Liu, J.; Chen, Y.; Huang, X.; Li, J.; Min, G. GNN-based long and short term preference modeling for next-location prediction. *Inf. Sci.* **2023**, *629*, 1–14. [\[CrossRef\]](#)
39. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
40. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [\[CrossRef\]](#)
41. Wang, B.; Shen, T.; Long, G.; Zhou, T.; Wang, Y.; Chang, Y. Structure-Augmented Text Representation Learning for Efficient Knowledge Graph Completion. In Proceedings of the WWW '21: The Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 1737–1748. [\[CrossRef\]](#)
42. Lv, X.; Lin, Y.; Cao, Y.; Hou, L.; Li, J.; Liu, Z.; Li, P.; Zhou, J. Do Pre-trained Models Benefit Knowledge Graph Completion? A Reliable Evaluation and a Reasonable Approach. In Proceedings of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 3570–3581. [\[CrossRef\]](#)
43. Wang, L.; Zhao, W.; Wei, Z.; Liu, J. SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models. In Proceedings of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 4281–4294. [\[CrossRef\]](#)
44. Youn, J.; Tagkopoulos, I. KGLM: Integrating Knowledge Graph Structure in Language Models for Link Prediction. In Proceedings of the 12th Joint Conference on Lexical and Computational Semantics, Toronto, ON, Canada, 13–14 July 2023; pp. 217–224. [\[CrossRef\]](#)
45. Lacroix, T.; Usunier, N.; Obozinski, G. Canonical Tensor Decomposition for Knowledge Base Completion. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 2869–2878.

46. Toutanova, K.; Chen, D. Observed versus latent features for knowledge base and text inference. In Proceedings of the Continuous Vector Space Models and their Compositionality, Beijing, China, 31 July 2015; pp. 57–66. [\[CrossRef\]](#)
47. Bollacker, K.D.; Evans, C.; Paritosh, P.K.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250. [\[CrossRef\]](#)
48. Safavi, T.; Koutra, D. CoDEX: A Comprehensive Knowledge Graph Completion Benchmark. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 8328–8350. [\[CrossRef\]](#)
49. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8024–8035.
50. Broscheit, S.; Ruffinelli, D.; Kochsiek, A.; Betz, P.; Gemulla, R. LibKGE—A knowledge graph embedding library for reproducible research. In Proceedings of the Empirical Methods in Natural Language Processing, Online, 16–20 November 2020; pp. 165–174. [\[CrossRef\]](#)
51. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.