Journal of Nonlinear and Convex Analysis
Volume 26, Number 6, 2025, 1691–1705

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

QUATERNIONIC KNOWLEDGE GRAPH EMBEDDING BASED ON RELATIONAL MAPPING

DELONG CUI, XIAOWEN CHEN, SHUANGYUAN LI, ZHIPING PENG*, QIRUI LI, JIEGUANG HE, JIANBIN XIONG, AND MINGTAO ZHENG

Abstract. Knowledge graph is an important branch of artificial intelligence that plays a vital role in multiple fields such as knowledge retrieval, intelligent question answering, and biomedical research. Hence, improving the completeness of knowledge graphs has become a research focus. However, most methods used in the existing studies on knowledge graphs capture specific relationship patterns and mapping attributes by designing specific vector spaces and operations. However, effective modeling of both relationship patterns (symmetry, antisymmetry, inversion, and composition) and relationship mapping properties (RMPs, including 1:1, 1:N, N:1, and N:N relationships) simultaneously is difficult. Hence, this study aimed to propose a quaternion knowledge graph embedding method based on relationship mapping. The rotation of quaternions enabled superior relationship pattern modeling capabilities using the projection to handle complex relationship mapping attributes. The experimental verification revealed the good performance of our method under different relationship patterns and mapping attributes.

1. Introduction

The concept of knowledge graph (KG) was officially proposed by Google in 2012, which is essentially a structured semantic network used to build a knowledge base to describe various real-world entities and their relationships [1]. The core components of this network are entities and their associated attribute value pairs, as well as the triple entity relationship entity connecting entities. A KG is used to establish a network structure of relationships between entities. This technology has been widely applied in various fields. However, existing KGs suffer from issues such as the lack of a large number of factual triples. This has led to the emergence of KG completion techniques, which predict and fill in the missing parts of triples based on known facts, thereby enhancing the integrity of the KG. The continuous emergence and development of KGs strongly support the field of artificial intelligence (AI) and are expected to promote innovation and development in various fields.

2020 *Mathematics Subject Classification.* 68T07, 68U10.

Key words and phrases. Knowledge graph completion, link prediction, quaternions, relational patterns.

The work presented in this paper was supported by: National Natural Science Foundation of China (62273109); Key Realm R&D Program of Guangdong Province (2021B0707010003); Guangdong Basic and Applied Basic Research Foundation (2022A1515012022, 2023A1515240020, 2023A 1515011913, 2024A1515012090); Key Field Special Project of Department of Education of Guangdong Province (2024ZDZX1034); Maoming Science and Technology Project (210429094551175, 2022DZXHT028, mmkj2020033); Projects of PhDs' Start-up Research of GDUPT (2023bsqd2012, 2023bsqd1002, 2023bsqd1013, XJ2022000301).

*Corresponding author.

Many problems in KG require adequate and valid triad data for their solution. Several methods have been proposed to assess the validity of a new triad. Among these, a common strategy is to apply the embedding model to calculate the score of a new triad for judging its validity. In this approach, a correct ternary usually achieves a higher score than an invalid ternary.

Embedding models are a crucial tool in this context. They map triplets to a continuous vector space, making these vectors semantically relevant. We can determine the similarity in entities by measuring the distance between them in vector space, which is crucial for evaluating the effectiveness of new triplets. Most jobs involve designing specific vector spaces and operations to model relational patterns and relationship mapping properties (RMP) [9]. For example, TransE embeds entities and relationships into the same vector space, treating relationships as translations, thus making it difficult to model symmetric relationships and RMPs. RotateE represents relationships as rotations on a complex plane to model four relationship patterns, but it cannot handle RMP due to the distance-preserving nature of rotations. RESCAL [15], DistMult [21], ComplEx [17], SimplE [7], TuckER [2], and DistMult have changed the relationship of bilinear models with diagonal matrices, but lack the ability to model asymmetric relationships. Neural network models lack clear geometric explanations. Rotate3D [6] and QuatE [25] introduce quaternion to extend rotation to both three-dimensional (3D) and four-dimensional (4D) spaces, achieving better performance with larger model capacity. Although Rotate3D and QuatE can capture the modeling ability of RMP well, they lack modeling combination relationships. In addition, some methods based on quaternion embedding also lack the ability to model two relationship patterns simultaneously, such as the introduction of special orthogonal groups [8] and hierarchies [22]. Moreover, certain combinations of quaternion and convolutional neural networks exist, but these methods lack a clear geometric explanation for modeling relational patterns and RMPs [24]. A projection-based approach was proposed in this study to simultaneously model relationship patterns and RMP based on quaternion embedding, aiming to address the lack of simultaneous modeling of relationship patterns and RMP, which improved the performance of the model under different relationship patterns and mapping attributes.

Currently, no method can simultaneously model all relationship patterns and RMPs. This is especially because existing predictions are irreversible transformations, resulting in modeling failures of inversion and combination patterns and suboptimal performance. Therefore, we proposed a quaternion knowledge graph embedding (KGE) based on relational mapping to solve this problem. We introduced a new reversible projection by modifying the matrix of ordinary household heads to address the limitations of existing models. We used projection and quaternion rotation to handle complex relational mapping attributes and achieve superior relational pattern modeling capabilities. This method achieved good results under different relationship patterns and mapping attributes.

2. Related work

A KG comprises several important relationship patterns and mapping attributes that describe different ways of connecting and mapping entities (Fig. 1):

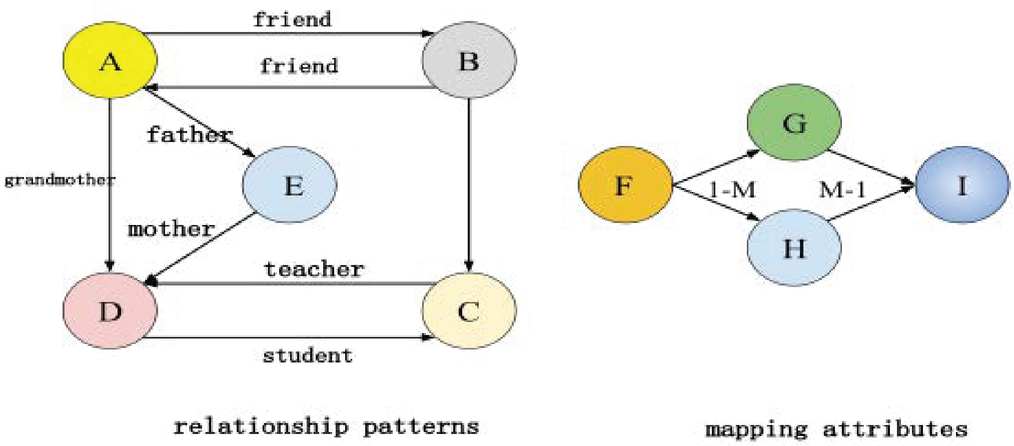

FIGURE 1. Relationship schema and mapping attributes

(1) Symmetric relationship: A friendship relation is an example of a symmetric relationship, where if A is a friend of B, then B is also a friend of A.

(2) Asymmetric relationships: The relationship between mother and child is an example of an asymmetric relationship, where the mother is a role of the child but the child is not a role of the mother.

(3) Mutual inverse relationship: The relationship between a teacher and a student is a mutual inverse relationship between the two, where the teacher teaches the student and the student is taught by the teacher.

(4) Combination: For example, " is_grandmother_of" is a combination of " is_father_of" and " is_mother_of".

The mapping properties of the relationship: 1:1 relationship: A unique mapping relationship exists between one entity and another. For example, each person has a unique ID number.

1:N relationship: One entity can have a mapping relationship with many different entities. For example, a teacher can teach more than one student.

N:1 relationship: Many different entities can be mapped to the same entity. For example, multiple students may have the same teacher.

N:N relationships: Many-to-many mapping relationships exist between multiple entities. For example, multiple students can be enrolled in multiple courses, and multiple students can be enrolled in a course. Figure 1 briefly describes the important relational patterns and mapping attributes in KG. Modeling and understanding relational patterns in KGs are extremely important for extracting meaningful information and knowledge from data. One of the key challenges in KGE is the modeling of relational patterns (e.g., symmetry, antisymmetry, inversion, and composition) and RMPs. Many KG complementation models have been proposed in recent years. These models are mainly categorized into distance translation-based, bilinear and tensor, complex vector, and neural network models. However, most of the work involves designing specific vector spaces and modeling these two relationship patterns through operations.

Translation model based on distance translation: This involves treating relationships in triplets as translations between head and tail entities, where entities and relationships can be mapped to each other. For example, TransE [3] represents relationships as translations but fails in symmetric and RMP modeling. Under the TransE framework, transformation embedding models based on relationship mapping attributes, such as TransC [12] and TransX, have been proposed. TransX represents many variants of TransE, such as TransH [19] and TransR [10]. TransR models entities and relationships in different spaces, but it lacks the ability to model inversion and combination relationships. Although the translation distance models use simple operations and limited parameters to learn embeddings, their embedding representations exhibit poor performance. For a relationship, TransH uses a joint representation of the hyperplane normal vector and the vector within the hyperplane. The head and tail entities should conform to the minimum triplet distance in the hyperplane, which can solve the 1:N problem because the projections are the same.

Bilinear and complex vector models: Tensor models use tensors to represent entities and relationships, and use tensor decomposition methods to construct mapping relationships between entities and relationships, such as RESCAL, DistMult, ComplEx, SimplE, and TuckER. DistMult changes the relationships of bilinear models to diagonal matrices, but it lacks the ability to model asymmetric relationships. ComplEx introduces complex embeddings to better model complex relationships in KGs, but it cannot model composition patterns. RotateE represents relationships as rotations on a complex plane to model four relationship patterns, but it cannot handle RMP due to the distance-preserving nature of rotations. Rotate3D and QuatE introduce quaternions for embedding representation of triples, extending rotation to both 3D and 4D spaces to achieve better performance with larger model capacity. DualE's improvement over the QuatE model extends KGE from quaternion space to dual quaternion space, but DualE [14] cannot capture relationship mapping attributes. Although QuatRE [4] enhances the correlation between head and tail entities based on quaternions, it still cannot model inversion and combination relationships well.

Neural network models: KG complementation models based on neural network models can automatically extract features and use the extracted features for entity or relationship prediction, such as R-GCN [16], ConvE [5], Conv-TransE [20], ConvKB [13], and so forth. R-GCN introduces a neural network of graphs as a graph encoder. ConvE uses convolutional operations to facilitate score computation. However, these methods lack an explicit geometric interpretation for modeling relational patterns and RMP. Detailed description as in the Table 1.

The effectiveness of KGE largely depends on the ability to model intrinsic relationship patterns and map attributes. However, the existing methods can only capture part of the method with insufficient modeling functions. They cannot fully

TABLE 1. Ability of classical models to model relational schemas and relationship mapping

Model	Asymmetrical	Asymmetric	Invert	Combinatorial	RMP
TransE	×	✓	✓	✓	×
TransX	✓	✓	×	×	✓
TransR	✓	✓	×	×	✓
DistMult	✓	×	×	×	✓
ComplEx	✓	✓	✓	×	✓
RotatE	✓	✓	✓	✓	×
Rotte3D	✓	✓	✓	×	✓
QuatE	✓	✓	✓	×	✓
DualE	✓	✓	✓	×	✓
QuatRE	✓	✓	✓	×	✓

model the intrinsic relationship patterns and imaging properties simultaneously. Therefore, we proposed a relational mapping quaternion embedding involving: (1) rotation of quaternions to achieve better possibilities for modeling relational patterns and (2) relational imaging to handle complex relational imaging properties. Modeling important relational patterns and assignment properties simultaneously is theoretically possible. The experimental results showed enhanced performance of our proposed model in completing KGs on known benchmark datasets.

KGE is crucial for effective information extraction and processing. However, existing methods have limitations in capturing intrinsic relationship patterns and mapping attributes. Therefore, we proposed a novel relational mapping embedding model using the concept of quaternions in nature and employing different mapping strategies. By mapping entities and relationships into quaternion space, we could better understand their relationships and better solve the problem of polysemy between entities and relationships. Compared with traditional methods, our model exhibited better pattern modeling ability and stronger mapping attribute processing ability. The experimental results showed that our model achieved good results on various benchmark datasets.

3. Quaternionic KGE based on relational mapping

3.1. **Quaternion.** Quaternion is a hypercomplex number composed of four real numbers. The commonly used symbols for quaternions are $q{=}q_r{+}q_i^i{+}q_j^j{+}q_k^k$, where $q_r, q_i, q_j, q_k \in R^n, i, j, k$ Virtual unit $ijk = i^2 = j^2 = k^2 = -1$.

Quaternions have specific multiplication rules, known as Hamiltonian multiplication. According to these rules, the result of quaternion multiplication can be obtained by multiplying each quaternion unit with each part of another quaternion and then performing the corresponding addition operation. Conjugation: The conjugation of quaternions Q is defined as: $q^2 = q_r - q_i^i - q_j^j - q_k^k$.

The quaternion q^{Δ} normalization is calculated as:

$$q^{\Delta} = \frac{q_r + q_i^i + q_j^j + q_k^k}{\sqrt{q_r^2 + q_i^2 + q_k^2}}. \tag{3.1}$$

Hamilton product: The Hamilton product of vectors and is calculated as:

$$\begin{aligned} q \otimes p = &\frac{1}{2}(q_r \bullet p_r - q_i \bullet p_i - q_j \bullet p_j - q_k \bullet p_k) \\ &+ (q_i \bullet p_r - q_r \bullet p_i - q_k \bullet p_j - q_j \bullet p_k)i \\ &+ (q_j \bullet p_r - q_k \bullet p_i - q_k \bullet p_j - q_j \bullet p_k)j \\ &+ (q_j \bullet p_r - q_k \bullet p_i - q_r \bullet p_j - q_i \bullet p_j)k \end{aligned} \tag{3.2}$$

where $\bullet$ denotes the product at the element level. The Hamiltonian product is not exchangeable, that is, $q \otimes p! = p \otimes q$. It determines another quaternion. A spatial rotation can be modeled using a quaternionic Hamiltonian product. Multiplying one quaternion p by another quaternion q allows q to be multiplied by the size of p. Then, a special type of rotation is performed in four dimensions. Thus, we can also rewrite the aforementioned equation as:

$$p \otimes q = p \otimes |q| \left(\frac{q}{p}\right). \tag{3.3}$$

Inner product: The four inner products of quaternion vectors p and q are obtained by the inner product between the corresponding scalar and imaginary components. The inner product is calculated as:

$$q \bullet p = q_r^T p_r + q_i^T p_i + q_j^T p_j + q_T^T p_k. \tag{3.4}$$

Quaternion embedding represents a step beyond traditional complex embeddings by introducing richer hypercomplex representations to more accurately portray the embedding of entities and relationships in a KG. More specifically, it refers to a hypercomplex embedding using three imaginary partial quantities; this approach is used to represent entities in a KG. At the same time, these relationships are also incorporated into the quaternion space to better model and describe them. The central idea here is to use the rotation operation of quaternions to represent these relationships. The quaternion embedding provides a higher-dimensional representation by introducing additional imaginary partial quantities, thus helping capture the associations between entities more accurately and making the embedding richer. Quaternion embeddings have greater expressive power compared with traditional complex representations because they can contain more information, which is especially significant when dealing with diversity relationships in KGs. The other advantages are as follows: (1) The use of Hamiltonian product properly captures potential interdependencies (between all components) and encourages more compact interactions between entities and relations. (2) Quaternions can express rotations in four dimensions with more degrees of freedom than rotations on complex planes. (3) The proposed framework is a generalization of the hypercomplexity space. Also, a number of projection operations are proposed to deal with the complexity of the RMP, and relational projection enables the KGE model to generate a specific relational representation for each entity.

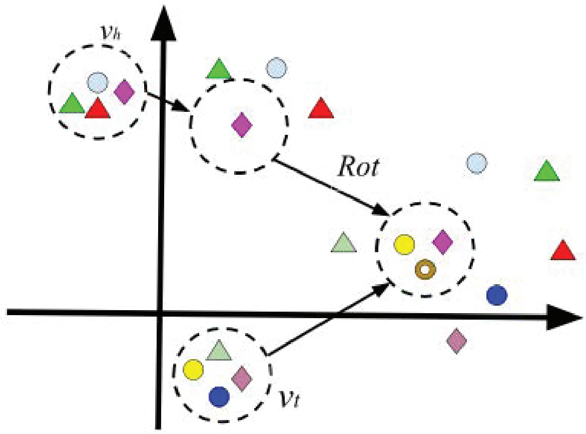

FIGURE 2. Projective mapping and rotation of head and tail entities

3.2. **General framework.** The model framework proposed in this study is shown in Figure 2. First, we represented the entities and relationships of triples (h, r, t_1) and (h, r, t_2) in the form of quadruple embeddings, that is $h, r, t_1, t_2 \in H_n$. Next, we introduced a new relationship mapping method that changed the relative distance between entities by increasing the distance between positive samples (diamonds) and negative samples (circles and triangles), besides reducing the distance between the tail entities t_1 and t_2 of the two positive samples. This method endowed the model with the ability to capture mapping attributes. Subsequently, we used relationship rotation to model relationship attributes and achieved the goal of mapping the projection head embedding h to the projection tail embedding.

This study mainly introduced Householder reflection transformation as a quaternionic mapping. A Householder matrix is often used in numerical algebra, such as orthogonal decomposition, and so forth. It is used to describe a basic reflection transformation over the origin hyperplane. Given a Householder matrix of the unit vector $u \in R^k, k \times k$, it is defined as:

$$H(u) = I - 2uu^T \tag{3.5}$$

where I is the unit matrix. Geometrically, a Householder matrix can provide a mirror reflection of the superplane of the unit normal vector.

Rotation transformation can effectively model four relationships, as shown in Table 1. However, because of its strict distance properties, simple relationship rotation, effective replication RMPs, and the introduction of various model relationship projection transformations, the model loses the ability to represent inverse relationships and combined modeling. However, these projection transformations are irreversible, resulting in the loss of inverse relationship and combined modeling abilities. To address this problem, this study introduced a new reversible projection called the Householder projection.

More specifically, given a unit variable $p \in R^k$ and a real scalar τ, the definition of the original Householder matrix was slightly modified, and the resulting matrix M, generated by the variables $p \in R^k$ and variables τ, was defined as:

$$M(p, \tau) = I - \tau p p^{\mathrm{T}}. \tag{3.6}$$

3.3. **Quaternion embedding.** The KGE model embedded triples into a low-dimensional space and defined a scoring function to measure the rationality of the triples. The number of effective triples was higher than the number of invalid triples. In the quaternion-embedded representation of a KG, entities and relationships are typically mapped to the quaternion space. The quaternion embeddings v_h, v_r, $v_t \in H_n$ of h, r, and t are represented as:

$$\begin{aligned} \nu_h &= \nu_{h,r} + \nu_{h,i} i + \nu_{h,j} j + \nu_{h,k} k, \\ \nu_r &= \nu_{r,r} + \nu_{r,i} i + \nu_{r,j} j + \nu_{r,k} k, \\ \nu_t &= \nu_{t,r} + \nu_{t,i} i + \nu_{t,j} j + \nu_{t,k} k. \end{aligned} \tag{3.7}$$

Among these, $\nu_{h,r}, \nu_{h,i}, \nu_{h,j}, \nu_{h,k}, \nu_{r,r}, \nu_{r,i}, \nu_{r,j}, \nu_{r,k}, \nu_{t,r}, \nu_{t,i}, \nu_{t,j}, \nu_{t,k} \in \mathbb{R}^n$ further associate each relationship with two quaternion vectors as:

$$\begin{aligned} \nu_h r &= \nu_{hr,r} + \nu_{hr,i} i + \nu_{hr,j} j + \nu_{hr,k} k, \\ \nu_t r &= \nu_{tr,r} + \nu_{tr,i} i + \nu_{tr,j} j + \nu_{tr,k} k. \end{aligned} \tag{3.8}$$

3.4. **Relational projection.** Given that the head and tail entities of a relationship typically have different implicit types to solve complex RMPs, some projection operations have been proposed and their effectiveness has been demonstrated through experiments. These relational projection operations provide a method for embedding KGs into models, generating specific relational representations for each entity. However, existing projection methods are often irreversible, leading to difficulties in modeling inversion and combination relationships. Therefore, based on quaternion embedding, we proposed a new projection method called relational projection, which allowed each relation r to project the head and tail entities using two independent sets of projection parameters. This method enabled the KGE model to generate specific relationship representations for each entity, thereby solving the problem of semantic diversity. The projection formula was as follows:

$$\hat{q} = q - \tau \langle q, p \rangle p \tag{3.9}$$

where determines the position of q on the p-axis and is a scalar. q, p is the dot product denoting the entities and relations represented by quaternion embeddings, respectively. The and relations are then used to project the head entity and tail entity to obtain and, respectively, as shown in the following equations:

$$\begin{aligned} h^{\triangleright} &= \mathrm{v}_h - \tau \langle \mathrm{v}_h, v_{hr} \rangle v_{hr}, \\ t^{\triangleright} &= \mathrm{v}_t - \tau \langle \mathrm{v}_t, v_{tr} \rangle v_{tr}. \end{aligned} \tag{3.10}$$

Next, the normalized was also used to rotate through the Hamiltonian product and then the quaternion inner product of was used to generate the ternary scores, which shared the quaternion components of the input vectors during the computation of the Hamiltonian product. The model in this study was to first use the two mappings in Eq. (3.8) for the head and tail entities, respectively. We defined the scoring function for the ternary (h, r, t) as:

$$f(h,r,t) = ((\nu_h - \langle \nu_h, \nu_{hr} \rangle \nu_{hr}) \otimes \nu_r) \bullet (\nu_t - \langle \nu_t, \nu_{tr} \rangle \nu_{hr}). \tag{3.11}$$

Loss function: We used the Adagrad optimizer to train the proposed model. We minimized the following loss function and regularized the model parameters θ as follows:

$$\begin{aligned} L &= \sum_{(h,r,t)\in(G\cup G')} \log\left(1+\exp\left(-l_{(h,r,t)} \bullet f_{(h,r,t)}\right)\right) + \lambda \|\theta\|_2^2 \\ &\text{in which } l_{(h,r,t)} = \begin{cases} 1 & for(h,r,t)\in G \\ 1 & for(h,r,t)\in G' \end{cases}. \end{aligned} \tag{3.12}$$

Among these, we used the $l2$ regularization rate; G represents the set of correct triples and invalid triples, and G' represents invalid triples. Algorithm 1 shows the KG complementation process using our proposed framework.

Algorithm 1: Embedding quaternion knowledge graph based on relation mapping

Input: *Entity set E, relation set R, and triple set T*
Output: *Missed triples T*
Process:
1. For e_i in E do: entityDict $\leftarrow e_i$; end for //Build entity dictionary
2. For r_i in R do: relationDict $\leftarrow r_i$; end for //Build relation dictionary
3. For $triple_i$ in T do: tripleDict $\leftarrow triple_i$; end for //Build triple dictionary
4. $V_{entity} \leftarrow$ entityDict //Representing entities with quaternions
5. $V_{relation} \leftarrow$ relationDict //Representing relations with quaternions
6. $V_{hr}, V_{ht} \leftarrow$ relationDict //Associate two quaternion vectors with each relationship
7. for $q = 1, 2, ..., Q$ in epoch do
8. for i in batch do:
9. $V_h, V_r, V_t \leftarrow$ entityDict, relationDict, tripleDict;
10. $h^{\triangleright}, t^{\triangleright} \leftarrow h, t$; /Entity mapping
11. $V_{h^{\triangleright}t} \overset{\otimes}{\leftarrow} h^{\triangleright}, V_r$ //Rotate the head entities
12. $\psi(h,r,t) \leftarrow dot_product(V_{h^{\triangleright}t}), t^{\triangleright}$; //quaternion inner product
13. loss $\leftarrow (t, \psi(h,r,t))$; //Compute loss
14. Minimize loss;
15. Update parameters;
16. end for
17. end for

We abstracted the KG completion task as a prediction task, such as predicting tail entities based on given head entities and relationships or predicting relationships based on head and tail entities. Therefore, we sorted the triples in the test set and used the scores generated by the score function to calculate the results.

We evaluated four well-known benchmark datasets:WN18 [23], FB15k [18], WN18RR [11], and FB15k-237. The relationship patterns of WN18 included symmetric, asymmetric, and inverted relationships. WN18RR removed the reverse relationship and retained the relationship patterns as symmetric, asymmetric, and combinatorial relationships due to the tendency of reverse relationships to cause leakage of the reverse relationship set. Similarly, the FB15K dataset contained symmetric, asymmetric, and inverse relationships. FB15K-237 included symmetric, anti-symmetric, and combinatorial relationships, with the reversal relationship removed. The characteristics of these four datasets are shown in Table 2.

Table 2. Statistics of datasets

Model	Entity quantity	Relation	Training set	Validation set	Testing set
WN18RR	409,43	11	86,835	3,034	3,134
FB15K-237	14,541	237	272,115	17,535	20,466
WN18	40,943	18	141,442	5,000	5,000
FB15K	14,951	1345	483,142	50,000	59,071

MRR, MR, Hits@1, and Hits@10 are the evaluation indicators used to evaluate experimental results. MRR measured the average reciprocal ranking of correct entities in the ranking function. Hits@k was the proportion of three effective rankings in the first k predictions among the four evaluation indicators. Lower MR, higher MRR, and higher Hits@k indicated better performance.

Training plan: We set up 100 batches for all datasets. The learning rate values were 0.02, 0.05, 0.1, with negative triple samples 1, 5, 10 for each training triple, embedding dimensions 128, 256, 384, and regularization rates of 0.05, 0.1, 0.2, 0.5. We trained 8000 times on WN18 and WN18RR and 2000 times on FB15k and FB15k-237. We monitored the scores of WN18 and WN18RR with an interval of 400 epochs, as well as the scores of FB15k and FB15k-237 after every 200 iterations on Hits@10 Score. We used grid search to select hyperparameters.

3.5. **Experimental results.** As shown in Tables 3 and 4, the model proposed in this study achieved good results on all four datasets. It outperformed the QuatE model in all metrics on the WN18RR dataset. On the FB15k-237 dataset, except for MR, all other metrics were superior to those of the QuatE model. In terms of metrics MRR and Hit@1, the improvement rates were 4.8% and 8.0%, respectively. On the WN18 dataset, the performance of MR and H@10 was superior to that of the QuatE model. Similarly, on the FB15k dataset, two metrics outperformed the QuatE model. The aforementioned experimental results demonstrated the effectiveness of the modeling method proposed in this study.

Our proposed model was compared with QuatE on the dataset FB15k-237 to further validate our effective modeling of relationship mapping attributes. The specific results are shown in Figures 3 and Figures 4. Our proposed model exhibited superiority under various mapping attributes. The improvement in our model

TABLE 3. Experimental results for WN18RR and FB15k-237

Model	WN18RR				FB15k-237			
	MRR	MR	H@10	H@1	MRR	MR	H@10	H@1
DistMult	.430	5110	.49	.39	.241	254	.419	.155
Complex	.440	5261	.51	.41	.247	339	.428	.158
R-GCN	-	-	-	.151	.248	-	.417	.151
KBGAN	.214	-	.472	-	.278	-	.458	-
ConvTransE	.46	-	.52	.43	.33	-	.51	.24
ConvE	.43	4187	.52	.40	.325	244	.501	.237
InteractE	.463	5202	.528	.430	.354	172	.535	.263
RotatE	.470	3277	.565	.422	.297	185	.480	.205
MuRR	.481	-	.566	.440	.335	-	.518	.243
Rot-Pro	.457	2815	.577	.397	.344	201	.540	.246
QuatE1	.481	3472	.564	.436	.311	176	.495	.221
QuatE2	.488	2341	.582	.438	.348	**87**	.550	.248
DualE	.482	-	.561	**.440**	.344	-	.518	.237
Ours	**.493**	**2042**	**.591**	**.440**	**.365**	88	**.559**	**.268**

TABLE 4. Experimental results for WN18RR and FB15k-237

Model	WN18				FB15k			
	MRR	MR	H@10	H@1	MRR	MR	H@10	H@1
DistMult	.797	655	.946	-	.798	42	.893	-
Complex	.941	-	.947	.936	.692	-	.840	.599
TorusE	.947	-	.954	.943	.733	-	.832	.674
ConvE	.943	374	.956	.935	.657	51	.831	.558
RotatE	.947	184	**.961**	.938	.699	32	.872	.585
QuatE1	.949	388	.960	.941	.770	41	.878	.700
QuatE2	.950	162	.959	**.945**	.782	17	**.900**	.711
DualE	**.953**	-	**.961**	**.945**	.790	-	.881	.734
Ours	.937	**105**	**.961**	.920	**.799**	32	892	**.738**

performance was more significant compared with QuatE, especially when dealing with complex N:1 relationships (prediction head entities) and N:N relationships. This performance improvement was due to our effective modeling of relationship mapping attributes, especially by capturing the complexity between relationships through the network.

This study evaluated the performance of the model under each relationship in the WN18RR dataset, as shown in Table 5, to further validate the performance of the model from a fine-grained perspective. The proposed model exhibited better performance in most relationships compared with the two representative rotation models RotatE and QuatE. Among these relationships, Hypernym/Instance Hypernym and Member Meronym/Has Part were the most common representatives

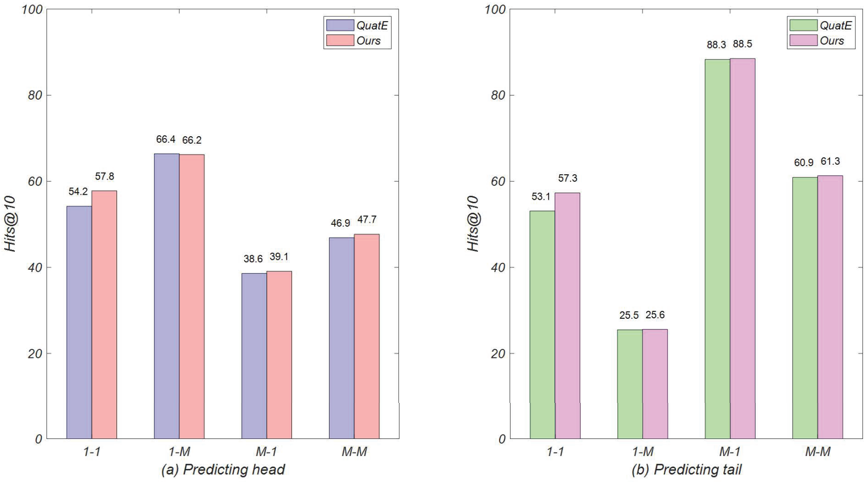

FIGURE 3. Performance of Hits@10 on the FB15k-237 dataset for testing each mapping relationship

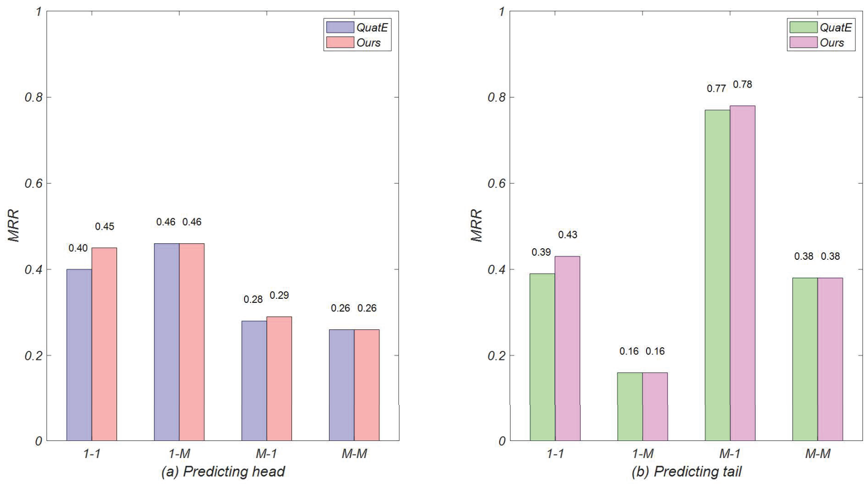

FIGURE 4. MRR's performance in testing each mapping relation on the FB15k-237 dataset

of inversion and combination relationships. Compared with QuatE, the method proposed in this study improved the Hypernym/Instance Hypernym and Has Part relationships by 10.4%, 4.9%, and 11.4%, respectively. This proved that relational mapping could better model inversion and combinatorial relationships. In addition, the model proposed in this study applied to the 1:N relationship member of the domain region and improved by 39.9% compared with RotatE, verifying superior modeling capabilities. Better performance could be achieved through relational mapping.

TABLE 5. MRR scores for the WN18RR test set for each relationship

Relationship	RotatE	QuatE	Ours
Hypernym	0.154	0.172	**0.190**
Instance_hypernym	0.324	0.362	**0.380**
member_meronym	0.255	0.236	0.236
synset_domain_topic_of	0.334	0.395	**0.500**
has_part	0.205	0.210	**0.234**
member_of_domain_usage	0.277	0.372	**0.487**
member_of_domain_region	0.243	0.140	**0.340**
derivationally_related_from	0.957	0.952	0.950
also_see	0.627	0.607	0.622
Similar_to	1.000	1.000	**1.000**
verb_group	0.968	0.930	0.921

4. CONCLUSIONS

In this study, we learned the embedding of entities and relations in quaternion space using the quaternion KGE method for relational mapping. The quaternions were rotated to achieve superior relational schema modeling capabilities, and projections were used to handle complex relational mapping attributes. Our proposed model displayed better performance for the KG complementation task on well-known benchmark datasets.

REFERENCES

[1] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto and R. Buyya *Big data computing and clouds: Trends and future directions*, Journal of Parallel and Distributed Computing **79** (2015), 3–15.

[2] I. Balažević, C. Allen and T. M. Hospedales, *Tucker: Tensor factorization for knowledge graph completion*, in: 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 5185–51945.

[3] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, *Translating embeddings for modeling multi-relational data*, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, Curran Associates Inc, 2013, pp. 2787– 2795

[4] Z. Cao, Q. Xu, Z. Yang, X. Cao and Q. Huang, *Dual quaternion knowledge graph embeddings*, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence. AAAI Press, 2021, pp. 6894–6902.

[5] T. Dettmers, P. Minervini, P. Stenetorp and S. Riedel, *Convolutional 2D knowledge graph embeddings*, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI Press, 2018, Article No. 221, pp.1811–1818.

[6] C. Gao, C. Sun, L. Shan., L. Lin and M. Wang. *Rotate3d:Representing relations as rotations in three-dimensional space for knowledge graph embedding*, in: Proceedings of the 29th ACM International Conference on Information& Knowledge Management. Association for Computing Machinery. 2020, pp. 385–394.

[7] M. Kazemi, D. Poole, *Simple embedding for link prediction in knowledge graphs*, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Curran Associates Inc. 2018, pp. 4289–430

[8] T. Le, H. Tran and B. Le, *Knowledge graph embedding with the special orthogonal group in quaternion space for link prediction*, Knowledge-Based Systems **266** (2023): 337.

[9] R. Li, J. Zhao, C. Li, D. He, Y. Wang, Y. Liu, H. Sun, S. Wang, W. Deng, Y. Shen, X. Xie and Q. Zhang, *House: Knowledge graph embedding with householder parameterization*, in: International Conference on Machine Learning. PMLR, 2022, pp. 13209–13224.

[10] H. Lin, Y. Liu, W. Wang, Y. Yue and Z. Lin, *Learning entity and relation embeddings for knowledge resolution*, Procedia Computer Science **108** (2017), 345–354.

[11] H. Luo, E. Haihong, L. Tan, G. Zhou, T. Yao and K. Wan, *DHGE: Dual-view hyper-relational knowledge graph embedding for link prediction and entity typing*, in Proceedings of the 37th AAAI Conference on Artificial Intelligence. AAAI Press, 2023, pp. 6467–6474.

[12] X. Lv, L. Hou, J. Li and Z. Liu. *Differentiating concepts and instances for knowledge graph embedding*, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2018, pp. 1971–1979.

[13] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen and D. Phung, *A novel embedding model for knowledge base completion based on convolutional neural network*, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2018, pp. 327–333

[14] D. Q. Nguyen, T. Vu, T. D. Nguyen and D. Phung, *Quatre: Relation-aware quaternions for knowledge graph embeddings*, in: Companion Proceedings of the Web Conference. Association for Computing Machinery. 2022, pp. 189–192.

[15] M. Nickel, V. Tresp and H. P. Kriegel, *A three-way model for collective learning on multi-relational data*, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, Omnipress, 2011, pp. 3104482-3104584.

[16] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Berg, I. Titov and M. Welling, *Modeling relational data with graph convolutional networks*, in: The Semantic Web: 15th International Conference, Springer International Publishing, 2018,pp. 593–607.

[17] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, *Complex embeddings for simple link prediction*, in: 2016 International Conference on Machine Learning, 2016, pp. 2071– 208.

[18] D. Wang, Y. Chen and B. C. Grau, *Efficient embeddings of logical variables for query answering over incomplete knowledge graphs*, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence. AAAI Press, 2023, pp. 4652–4659.

[19] Z. Wang, J. Zhang, J. Feng and Z. Chen, *Knowledge graph embedding by translating on hyperplanes*, in: Proceedings of the 28th AAAI conference on artificial intelligence, AAAI Press, 2014, pp. 1112–1119.

[20] S. Yan, Z. Zhang, G. Xu, X. Sun, S. Li and S. Wang, *Modeling N-ary relational data as gyro-polygons with learnable gyro-centroid*, Knowledge-Based Systems **251** (2022): 109164.

[21] B. Yang, W. Yih, X. He, J. Gao and L. Deng, *Embedding entities and relations for learning and inference in knowledge bases*, arxiv preprint arxiv:1412.6575, 2014.

[22] J. Yang, X. Ying, Y. Shi, X. Tong, R. Wang, T. Chen and B. Xing, *Learning hierarchy-aware quaternion knowledge graph embeddings with representing relations as 3D rotations*, in: Proceedings of the 29th International Conference on Computational Linguistics. International Committee on Computational Linguistics, 2022, pp. 2011–2023.

[23] Z. Yao, W. Zhang, M. Chen, Y. Huang, Y. Yan and H. Chen, *Analogical inference enhanced knowledge graph embedding*, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence. AAAI Press, 2023, pp. 4801–4808.

[24] S. Zhang, X. Liang, H. Tang, X. Zheng, A. X. Zhang and Y. Ma, *DuCape: Dual quaternion and capsule network-based temporal knowledge graph embedding*, ACM Transactions on Knowledge Discovery from Data **17** (2023): Article No. 104.

[25] S. Zhang, Y. Tay, L. Yao and Q. Liu, *Quaternion knowledge graph embeddings*, in Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc, 2019, pp. 2735–2745.

Manuscript received March 14, 2024
revised September 12, 2024

D. L. Cui
Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Maoming, Guangdong, China;
School of Electronic Information Engineering, Guangdong University of Petrochemical Technology
E-mail address: delongcui@gdupt.edu.cn

X. W. Chen
School of Electronic Information Engineering, Guangdong University of Petrochemical Technology, Maoming, Guangdong, China
E-mail address: 1803463546@qq.com

S. Y. Li
Jilin Institute of Chemical Technology, Jilin, China
E-mail address: 767641@qq.com

Z. P. Peng
Jiangmen Polytechnic, Jiangmen, Guangdong, China
E-mail address: zhipingpeng@gdupt.edu.cn

Q. R. Li
Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Maoming, Guangdong, China;
School of Computer Science and Engineering, Guangdong University of Petrochemical Technology, Maoming, Guangdong, China
E-mail address: liqirui@gdupt.edu.cn

J. G. He
School of Computer Science and Engineering, Guangdong University of Petrochemical Technology, Maoming, Guangdong, China
E-mail address: jieguanghe@gdupt.edu.cn

J. B. Xiong
School of Automation, Guangdong Polytechnic Normal University, Guangdong, Guangzhou, China
E-mail address: 276158903@qq.com

M. T. Zheng
Maoming Modern Agricultural Research Institute, Maoming, Guangdong, China
E-mail address: mtzheng@scau.edu.cn