Learning Hierarchy-Aware Quaternion Knowledge Graph Embeddings with Representing Relations as 3D Rotations

Jinfa Yang, Xianghua Ying*, Yongjie Shi, Xin Tong, Ruibin Wang, Taiyan Chen, Bowei Xing

Key Laboratory of Machine Perception (MOE)
School of Intelligence Science and Technology, Peking University
{jinfayang, xhying, shiyongjie, xin_tong, robin_wang}@pku.edu.cn, chenty@stu.pku.edu.cn, 2017xbw@pku.edu.cn

Abstract

Knowledge graph embedding aims to represent entities and relations as low-dimensional vectors, which is an effective way for predicting missing links. It is crucial for knowledge graph embedding models to model and infer various relation patterns, such as symmetry/antisymmetry. However, many existing approaches fail to model semantic hierarchies, which are common in the real world. We propose a new model called HRQE, which represents entities as pure quaternions. The relational embedding consists of two parts: (a) Using unit quaternions to represent the rotation part in 3D space, where the head entities are rotated by the corresponding relations through Hamilton product. (b) Using scale parameters to constrain the modulus of entities to make them have hierarchical distributions. To the best of our knowledge, HRQE is the first model that can encode symmetry/antisymmetry, inversion, composition, multiple relation patterns and learn semantic hierarchies simultaneously. Experimental results demonstrate the effectiveness of HRQE against some of the SOTA methods on four well-established knowledge graph completion benchmarks.

1 Introduction

Knowledge graphs represent human knowledge of the real world as structured triples—(head entity, relation, tail entity) also known as (subject, predicate, object). There are some outstanding knowledge graphs, such as WordNet (Miller, 1995), Freebase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2015). They have gained widespread attention for their successful usage in various applications (e.g., question answering, natural language processing, and recommendation systems). Although millions of entities and billions of facts exist in large-scale knowledge graphs, they still suffer from the incompleteness problem. Therefore, knowledge graph completion (also known as link prediction) which aims to predict missing links among entities based on the known triples has gained growing interest. Learning low-dimensional representations of entities and relations for Knowledge graphs is an effective solution for this task.

Learning knowledge graph embeddings in the complex space \( \mathbb{C} \) or quaternion space \( \mathbb{H} \) has been proven to be a highly effective inductive bias, largely owing to their ability to model connectivity patterns of the relations. For example, ComplEx (Trouillon et al., 2016), which infers new relational triplets with the asymmetrical Hermitian product can model the symmetry/antisymmetry patterns. RotatE (Sun et al., 2019), which represents entities as points in a complex space and relations as rotations, can model relation patterns including symmetry/antisymmetry, inversion, and composition. DualE (Cao et al., 2021), which combines rotation and translation in dual quaternion space can additionally model the multiple relations pattern. However, many existing models fail to model semantic hierarchies in knowledge graphs.

Semantic hierarchy is a ubiquitous property in knowledge graphs. For instance, WordNet contains the triple [arbor/cassia/palm, hyponym, tree], where “tree” is at a higher level than “arbor/cassia/palm” in the hierarchy. Freebase contains the triple [America, /location/location/contains, California/Los Angeles], where “California/Los Angeles” is at a lower level than “America” in the hierarchy. Although there exists some work that takes the hierarchy structures into account (Xie et al., 2016; Zhang et al., 2020), they usually require additional data to obtain the hierarchy information or cannot model various relation patterns. Therefore, it is still challenging to find an approach that is capable of modeling the various relation patterns and semantic hierarchy simultaneously.
| Model               | Multiple | Symmetry | Relation Patterns | Antisymmetry | Inversion | Composition | Hierarchy-Aware |
|---------------------|----------|----------|-------------------|--------------|-----------|-------------|----------------|
| TransE (Bordes et al., 2013) | ✗        | ✗        | ✓                 | ✓            | ✓         | ✓           | ✗              |
| DistMult (Yang et al., 2015) | ✓        | ✓        | ✓                 | ✗            | ✗         | ✓           | ✗              |
| ComplEx (Trouillon et al., 2016) | ✗        | ✓        | ✓                 | ✓            | ✗         | ✓           | ✗              |
| RotateE (Sun et al., 2019) | ✓        | ✓        | ✓                 | ✓            | ✗         | ✓           | ✗              |
| QuatE (Zhang et al., 2019) | ✓        | ✓        | ✓                 | ✗            | ✓         | ✗           | ✗              |
| HAKE (Zhang et al., 2020) | ✓        | ✓        | ✓                 | ✓            | ✓         | ✗           | ✗              |
| DualE (Cao et al., 2021) | ✓        | ✓        | ✓                 | ✓            | ✓         | ✓           | ✓              |
| QuatRE (Nguyen et al., 2022) | ✓        | ✓        | ✓                 | ✓            | ✗         | ✗           | ✗              |
| RQE                 | ✓        | ✓        | ✓                 | ✓            | ✓         | ✓           | ✗              |
| HRQE                | ✓        | ✓        | ✓                 | ✓            | ✓         | ✓           | ✓              |

Table 1: The pattern modeling and hierarchy-aware abilities of several models

In this paper, we propose Rotation Based Quaternion Knowledge Graph Embeddings (RQE) and its Hierarchy-aware extension HRQE. More concretely, we represent entities as pure quaternions with three imaginary components i, j and k. The relational embedding consists of two parts: (a) Using unit quaternions to represent the rotation part in 3D space, where the head entities $Q_h$ are rotated by the corresponding relations through Hamilton product. (b) Using scale parameters to constrain the modulus of entities $Q_h$ and $Q_t$ to make them have hierarchical distributions.

To summarize, our contributions are as follows: 1) We propose a new framework called HRQE based on quaternion rotation. 2) To the best of our knowledge, HRQE is the first model that can encode symmetry/antisymmetry, inversion, composition, multiple relation patterns and learn semantic hierarchies simultaneously. 3) We conduct a series of theoretical and empirical analyses to show the strength of HRQE against some of the SOTA methods.

2 Related Work

2.1 Knowledge Graph Embedding Models

Roughly speaking, we can divide knowledge graph embedding models into translational distance models and semantic matching models. The former use distance-based score functions, while the latter use similarity-based ones.

**Translational Distance Models.** TransE (Bordes et al., 2013) is the most widely used translation distance constraint model. It assumes that entities and relations satisfy $head + relation \approx tail$. However, TransE cannot handle 1-1-N, N-1-1, and N-1-N relations well (Wang et al., 2014). TransH (Wang et al., 2014) is proposed to compensate for the shortcomings of TransE. It projects entities onto relation-specific hyperplanes. TransR (Lin et al., 2015) has a very similar idea to TransH, which introduces relation-specific spatial transformations instead of hyperplanes. TranSparse (Ji et al., 2016) simplifies TransR by forcing the projection matrix to be sparse. Moreover, RotatE (Sun et al., 2019) defines each relation as a rotation from the source entity to the target entity in a complex vector space, which can represent various relation patterns including symmetry/antisymmetry, inversion and composition.

**Semantic Matching Models.** RESCAL (Nickel et al., 2011) is a tensor factorization model which represents each relation as a full-rank matrix and obtains score function by matrix multiplication. DistMult (Yang et al., 2015) simplifies RESCAL by restricting relation matrices to be diagonal. However, Distmult assumes that all relations are symmetric. ComplEx (Trouillon et al., 2016) extends DistMult to complex space, and uses conjugate-transpose to model asymmetric relations. QuatE (Zhang et al., 2019) extends the complex space into the quaternion space with two rotating surfaces. DualE (Cao et al., 2021) combines rotation and translation in dual quaternion space. ConvE (Dettmers et al., 2018) and InteractE (Vashishth et al., 2020) employ convolutional neural networks to build score functions.

2.2 The Ways to Model Hierarchy Structures

Another related problem is how to model hierarchy structures in knowledge graphs. Xie et al. (2016) propose TKRL, which requires additional hierarchical type information for entities. Zhang et al. (2018) use clustering algorithms to model the hierarchical relation structures. Zhang et al.
(2020) proposed HAKE, which maps entities into the polar coordinate system for hierarchy-aware. Inspired by HAKE, we project entities into 3D space and constrain their rotations and modulus with corresponding relations. In addition to learning the semantic hierarchy, we can better encode various relation patterns such as multiple relations.

3 Quaternion Background

A quaternion \( Q \in \mathbb{H} \) is a hyper-complex number consisting of a real and three separate imaginary components (Hamilton, 1844), defined as \( Q = a + bi + cj + dk \), where \( a, b, c, d \in \mathbb{R} \) and \( i, j, k \) are imaginary units. \( i, j \) and \( k \) are satisfied with Hamilton’s rules (\( i^2 = j^2 = k^2 = \text{ijk} = -1 \)). And based on these rules, more non-commutative multiplication rules can be derived, such as \( ij = k, ji = -k, jk = i, kj = -i, ki = j, \) and \( ik = -j \).

Some widely used operations of quaternion algebra are introduced as follows:

- **Quaternion Conjugate**: The conjugate of a quaternion \( Q \) is defined as \( \overline{Q} = a - bi - cj - dk \).
- **Quaternion Norm**: The norm of a quaternion \( Q \) is defined as \( |Q| = \sqrt{a^2 + b^2 + c^2 + d^2} \).
- **Pure Quaternion**: A pure quaternion \( Q \in \mathbb{H}_p \) is defined as a quaternion whose scalar part is zero. Usually, we convert the 3D space point \((x, y, z)\) into a pure quaternion \( Q = 0 + x'i + y'j + z'k \), \( x, y \) and \( z \in \mathbb{R} \) for further quaternion operations.
- **Quaternion-Inner Product**: The quaternion inner product between \( Q_1 = a_1 + b_1i + c_1j + d_1k \) and \( Q_2 = a_2 + b_2i + c_2j + d_2k \) is obtained by taking the inner products between corresponding scalar and imaginary components and returns a scalar \( Q_1 \cdot Q_2 = \langle a_1, a_2 \rangle + \langle b_1, b_2 \rangle + \langle c_1, c_2 \rangle + \langle d_1, d_2 \rangle \).
- **Quaternion Multiplication (Hamilton Product)**: The quaternion multiplication is composed of all the standard multiplications of factors in quaternions and returns another quaternion, defined as:

\[
Q_1 \otimes Q_2 = (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)i + (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)j + (a_1d_2 - b_1c_2 - c_1b_2 + d_1a_2)k.
\]

**Quaternion Rotation**: If \( Q_r = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} u \), where \( u \in \mathbb{R}i + \mathbb{R}j + \mathbb{R}k \) is a unit vector, the result of pure quaternion \( Q = 0 + xi + yj + zk \) rotating \( \theta \) around the axis \( u \) is \( Q' = 0 + x'i + y'j + z'k \), then

\[
Q' = Q_r \otimes Q \otimes \overline{Q}_r.
\]

4 Method

In this section, we introduce our proposed model HRQE. First of all, we elaborate the details of our framework, which mainly consists of two parts: (1) rotate the head entity using the unit relation quaternion and score each triplet with inner product between the rotated head quaternion and the tail quaternion; (2) limit the norm of the head quaternion and the tail quaternion with the relation modulus part. After that, we provide a series of analyses to show the strength of our framework.

**Symbol Description.** Suppose that we have a knowledge graph \( G \) consisting of \( N \) entities and \( M \) relations. We formulate all the entity embeddings as a pure quaternion matrix \( Q \in \mathbb{H}_p^{N \times k} \), where each row is an embedding vector for a specific entity of dimensionality \( k \), and denote the relation embeddings as rotation part \( W \in \mathbb{H}_p^{M \times k} \) and modulus part \( w \in \mathbb{H}_p^{M \times k} \). Given a triplet \((h, r, t)\), the embedding of head entity \( h \) is denoted as \( Q_h = \{0 + x_hi + y_hj + z_hz : x_h, y_h, z_h \in \mathbb{R}^k\} \) and the embedding of the tail entity \( Q_t = \{0 + x_ti + y_tj + z_tz : x_t, y_t, z_t \in \mathbb{R}^k\} \), where \( Q_h, Q_t \in Q \).
Then we denote the relation \( r \) as rotation part \( W_r = \{ a_r + b_r i + c_r j + d_r k : a_r, b_r, c_r, d_r \in \mathbb{R}^k \} \) and modulus part \( w_r = \{ e_r : e_r \in \mathbb{R}^k \} \), where \( W_r \in W, w_r \in w \).

4.1 Hierarchy-Aware Rotation Quaternion Embeddings

The Rotation Part. We first normalize the relation quaternion \( W_r \) to a unit quaternion \( W_r^u \) to eliminate the scaling effect by dividing \( W_r \) by its norm:

\[
W_r^u = \frac{W_r}{|W_r|} = \frac{a_r + b_r i + c_r j + d_r k}{\sqrt{a_r^2 + b_r^2 + c_r^2 + d_r^2}}.
\] (3)

Secondly, we rotate the head entity \( Q_h \) by doing Hamilton product with \( W_r^u \) and \( W_r^q \):

\[
Q_h'(r_h', x_h', y_h', z_h') = W_r^q \otimes Q_h \otimes W_r^q,
\] (4)

where \( \otimes \) denotes the element-wise multiplication between two vectors. Then the rotation part scoring function \( \phi_r(h, r, t) \) is defined by the quaternion inner product:

\[
\phi_r(h, r, t) = Q_h \cdot Q_t = \langle x_h', x_t \rangle + \langle y_h', y_t \rangle + \langle z_h', z_t \rangle.
\] (5)

We separate the rotation part as an independent model RQE, which achieves impressive results (refer to Section 5).

The Modulus Part. As shown in Figure 1a, the rotation part allows the head entity to rotate in 3D space to approximate the tail entity. The modulo length of entities is used to represent the hierarchical distribution of entities. The modulus part of relations is used to measure the hierarchical difference between head and tail entities, which is beneficial for learning hierarchy-aware, see Section 4.2 for details. The modulus part scoring function \( \phi_m(h, r, t) \) is defined as:

\[
\phi_m(h,r,t) = -\|w_r|Q_h| - |Q_t||1
= -\|w_r\sqrt{x_h^2 + y_h^2 + z_h^2} - \sqrt{x_t^2 + y_t^2 + z_t^2}\|1.
\] (6)

Finally, The scoring function of HRQE is:

\[
\phi(h, r, t) = \phi_r(h, r, t) + \lambda \phi_m(h, r, t),
\] (7)

where \( \lambda \in \mathbb{R} \) is a parameter that learned by the model.

Loss Function. Following Trouillon et al. (2016), We formulate the task as a classification problem and adopt the cross-entropy loss as our loss function. \( \Omega \) and \( \Omega' = E \times R \times E - \Omega \) are used to denote the set of observed triplets and the set of unobserved triplets, respectively. Moreover, we use the \( \ell_2 \) norm with regularization rates \( \lambda_1 \) and \( \lambda_2 \) to regularize \( Q \) and \( W \):

\[
L = \sum_{r(h,t) \in \Omega', \Omega^-} \log(1 + \exp(-Y_{hrt} \phi(h, r, t)))
+ \lambda_1||Q||^2 + \lambda_2||W||^2,
\] (8)

where \( \Omega^- \subset \Omega' \) with negative sampling strategies such as uniform sampling, bernoulli sampling (Wang et al., 2014), and adversarial sampling (Sun et al., 2019). \( Y_{hrt} \in \{-1, 1\} \) represents the corresponding label of the triplet \((h, r, t)\). We optimize the loss function by utilizing Adagrad (Duchi et al., 2011).

4.2 Discussion

In this part, we discuss the theoretical properties of HRQE. We summarize several popular knowledge graph embedding models in Appendix A.5 and definitions of various relation patterns in Appendix A.1.

Capability in Modeling Multiple Relations. For multiple relations such as (A, classmate, B) and (A, neighbor, B) \( \in G \), HRQE allows multiple expressions for the relations when the head and tail entities are fixed. As shown in Figure 1b (here the modulus part is simplified as \( w_r = 1 \)), the red arc passes through \( r \) vertex \( P_r \) and the angle bisector vertex \( P_{0/2} \). \( r' \) with vertex on the red arc can also make \( h \) rotate to \( t \). That is, \( \exists r' \neq r, (h, r, t) \) and \((h, r', t) \in G \).

Capability in Modeling Symmetry/Antisymmetry, Inversion and Composition. The flexibility and representational power of quaternion rotation enable us to model various relation patterns at ease. Since HRQE degenerates to RQE when \( \lambda = 0 \), we mainly use RQE to discuss. When the rotation angle \( \theta = [0^\circ, 180^\circ, 270^\circ, 360^\circ] \), RQE can model the symmetry pattern, and when \( W_r^u = W_r^q \), RQE can model the inversion pattern. The specific lemmas and proofs are as follows:

Lemma 1 HRQE can infer the symmetry/antisymmetry pattern. (See proof in Appendix A.2)
### 5 Experiments and Results

#### 5.1 Experimental Setup

We evaluate our proposed models on four widely used benchmarks, which are statistically summarized in Table 4.

**Datasets:** WN18 (Bordes et al., 2013) is extracted from WordNet (Miller, 1995), a database featuring lexical relations and conceptual-semantic between words. The dataset has many inverse relations and the mainly relation patterns are symmetric/antisymmetric and inversion. WN18RR (Dettmers et al., 2018) is a subset of WN18, with inverse relations removed. The main relation patterns are symmetric/antisymmetric and composition. In WN18 and WN18RR, most of the triples consist of hyponym and hypernym relations which make them tend to follow a strictly hierarchical structure. FB15K (Bordes et al., 2013) is extracted from Freebase (Bollacker et al., 2008), which is a large-scale knowledge graph containing general human knowledge. The key of link prediction on FB15K is to model and infer the symmetry/antisymmetry and inversion patterns. FB15K-237 (Toutanova and Chen, 2015) is a subset of FB15K, with inverse relations removed. Therefore, the key to link prediction on FB15K-237 boils down to model and infer symmetry/antisymmetric and composition patterns.

**Evaluation Protocol:** For each triple \((h, r, t)\) in the test dataset, we replace either the head entity \(h\) or the tail entity \(t\) with the total list of the embedding entities. Then, we base the score function to rank the candidate entities in descending order. The filtered setting is used to remove some correct results that appear in the training set or valida-
### Table 3: Evaluation results on WN18RR, FB15k-237 datasets. The best scores are in bold, while the second best scores are in underline.

| Models       | WN18RR | FB15K-237 |
|--------------|--------|-----------|
|              | MR     | MRR(%)    | @10 | @3 | @1    | MR     | MRR(%)    | @10 | @3 | @1 |
| TransE       | 3384   | 22.6      | 50.1 | -  | -     | 357    | 29.4      | 46.5 | -  | -  |
| DistMult     | 5100   | 43.0      | 49   | 44 | 39    | 254    | 24.1      | 41.9 | 26.3 | 15.5 |
| ComplEx      | 5261   | 44.3      | 51   | 46 | 41    | 339    | 24.7      | 42.8 | 27.5 | 15.8 |
| ConvE        | 4187   | 43.0      | 52   | 44 | 40    | 244    | 32.5      | 50.1 | 35.6 | 23.7 |
| InteractE    | 5202   | 46.3      | 52.8 | -  | 43.0  | 172    | 35.4      | 53.5 | -  | 26.3 |
| RotatE       | 3277   | 47.0      | 56.5 | 48.8| 42.2  | 185    | 29.7      | 48.0 | 32.8 | 20.5 |
| a-RotatE     | 3340   | 47.6      | 57.1 | 49.2| 42.8  | 177    | 33.8      | 53.3 | 37.5 | 24.1 |
| QuaTE        | 2314   | 48.8      | 58.2 | 50.8| 43.8  | 87     | 34.8      | 55.0 | 38.2 | 24.8 |
| ComplEx-N3   | 5261   | 44.3      | 51   | 46 | 41    | 172    | 35.4      | 53.5 | -  | 26.3 |
| TuckER       |         |           |      |    |       |        |           |      |    |    |
| MURP         |         |           |      |    |       |        |           |      |    |    |
| RoTH         | 2293   | 49.1      | 58.6 | 51.1| 44.1  | 165    | 34.7      | 54.3 | 38.5 | 25.0 |
| Rotat3D      | 3328   | 48.9      | 57.9 | 50.5| 44.2  | 165    | 34.7      | 54.3 | 38.5 | 25.0 |
| HAKE         | 2270   | 49.2      | 58.4 | 51.3| 44.4  | 91     | 36.5      | 55.9 | 40.0 | 26.8 |
| DualE        | 1986   | 49.3      | 59.2 | 51.9| 43.9  | 88     | 36.7      | 56.3 | 40.4 | 26.9 |
| QuaRE        | 1198   | 50.5      | 60.1 | 52.4| 45.4  | 89     | 37.2      | 56.9 | 40.7 | 27.5 |

### Table 4: Number of entities, relations, and observed triplets in each split for benchmarks.

| Dataset   | #En | #Re | #train | #valid | #test |
|-----------|-----|-----|--------|--------|-------|
| WN18      | 40,943 | 18  | 141,442 | 5,000   | 5,000  |
| WN18RR    | 40,943 | 11  | 86,835  | 3,034  | 3,134  |
| FB15K     | 14,951 | 1,345 | 483,142 | 50,000 | 59,071 |
| FB15k-237 | 14,541 | 237  | 272,115 | 17,535 | 20,466 |

### Baselines:

- We compare HRQE with a number of strong baselines. For Translational Distance Models, we report TransE (Bordes et al., 2013), TorusE (Ebisu and Ichise, 2018), RotatE (Sun et al., 2019), Rotat3D (Gao et al., 2020), ROTH (Chami et al., 2020) and HAKE (Zhang et al., 2020); For Semantic Matching Models, we report DistMult (Yang et al., 2015), HolE (Nickel et al., 2016), ComplEx (Trouillon et al., 2016), ComplEx-N3 (Lacroix et al., 2018), SimplE (Kazemi and Poole, 2018), TuckER (Balažević et al., 2019), ConvE (Dettmers et al., 2018), R-GCN (Schlichtkrull et al., 2018), NKGE (ConvE based) (Wang et al., 2018), InteractE (Vashishth et al., 2020), QuaTE (Zhang et al., 2019), DualE (Cao et al., 2021), and QuaRE (Nguyen et al., 2022).

### Implementation Details:

- The best models are selected by early stopping on the validation set with Hits@10. The ranges of the hyperparameters for the grid search are set as follows: The embedding size $k$ is selected in $\{100, 200, 300, 400, 500\}$. The regularization rates $\lambda_1$ and $\lambda_2$ are adjusted in $\{0, 0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.5\}$. The learning rate is chosen from $\{0.01, 0.02, 0.05, 0.1\}$, the number of negative triples sampled per training triple is selected from $\{1, 2, 5, 10\}$. In addition, we create $\{10, 100\}$ batches of training samples for the different datasets. We report RQE and HRQE with type constraints (Krompaß et al., 2015). The training strategies of self-adversarial negative sampling (Sun et al., 2019) and N3 regularization with reciprocal learning (Lacroix et al., 2018) are not used for RQE and HRQE. All hyper-parameters of our models are provided in the appendix A.6, and our code is available at https://github.com/Jinfa/HRQE.

### 5.2 Results

The empirical results on four datasets are reported in Table 2 and Table 3. HRQE performs extremely competitively compared to the existing state-of-the-art models across all metrics. As a rotation-
Figure 2: Histograms of relation embeddings for different relation patterns. The corresponding relation is as follows: $r_1$ is similar_to; $r_2$ is has_part; $r_3$ is part_of; $r_4$ is /location/administrative_division/capital /location/administrative_division_capital_relationship/capital; $r_5$ is /location/hud_county_place/place; $r_6$ is base/areas/schema/administrative_area/capital.

Based model, HRQE outperforms the two representative rotation models RoatE, and Roatat3D. As a hierarchy-sensitive model, HRQE outperforms the representative hierarchy-aware model HAKE. Also, we outperform other quaternion-valued models such as QuatE, DualE, and QuatRE.

On the WN18 dataset, HRQE outperforms all the baselines on all metrics except Hit@1. HRQE achieves slightly lower results on H@1 than QuatE and RotatE, but surpasses them on the other four metrics, especially on MR with a 56% improvement over QuatE. RQE outperforms HRQE on the FB15K dataset, while the results of them on the validation set are close with 87%. We speculate that excessive inverse relations in FB15K affect the expression of HRQE hierarchy-aware modules. The other recent models a-RotatE, QuatE, and Rotat3D achieve comparable results.

As shown in Table 3, HRQE achieves the best performance over existing state-of-the-art models on the two datasets where trivial inverse relations are removed. On WN18RR in which there are a number of symmetry relations, TransE cannot learn the symmetric relation pattern, so it performs not well. In contrast, the rotation family can achieve good results, and HRQE has further refreshed the performance to achieve the optimal. In addition, HRQE’s performance on MR is also impressive, with a 48% improvement over QuatE. WN18 and WN18RR contain hyponym and hypernym relations which make them tend to follow a strictly hierarchical structure, and HRQE’s performance demonstrates its ability to learn hierarchically. On FB15K-237, HRQE also achieved better results compared with the previous state-of-the-art models, which shows that HRQE can learn the composition relation pattern better.

Table 5: Evaluation results of multiple relations on FB15k dataset.

| Models     | Prediction Head (MRR) | Prediction Tail (MRR) |
|------------|-----------------------|-----------------------|
| TransE     | 44.3                  | 48.4                  |
| RotatE     | 66.8                  | 79.4                  |
| RQE        | 67.8                  | 87.7                  |

5.3 Model Analysis

Analysis on Multiple Relations. In the test set of FB15K, 38121 are single-relation triples (1-1-1), and 20950 are multi-relation triples (1-N-1). To avoid the influence of the modulus part, we choose RQE as the comparison model. Table 5 shows that RQE can better deal with multi-relational triples than TransE and RotatE.

Visualize Some Typical Relation Patterns. To
Models                | Prediction Head (Hits@10) | Prediction Tail (Hits@10) |
|----------------------|--------------------------|--------------------------|
|                      | 1-1-1        | 1-1-N        | N-1-1        | N-1-N        | 1-1-1        | 1-1-N        | N-1-1        | N-1-N        |
| QuatE                | 54.2         | 66.4         | 38.6         | 46.9         | 53.1         | 25.5         | 88.3         | 60.9         |
| QuatRE               | 58.9         | 66.4         | 39.3         | 48.1         | 59.9         | 26.8         | 88.9         | 61.7         |
| HRQE                 | 63.5         | 66.4         | 42.5         | 48.7         | 63.0         | 28.1         | 88.9         | 62.2         |

Table 6: Evaluation results of complex relations on FB15k-237 dataset. The first two rows are taken from (Nguyen et al., 2022)

Further verify the learned relation patterns, we visualize some examples. For symmetry pattern, HRQE encode with rotation angle \( \theta = [0^\circ, 180^\circ, 360^\circ] \) (correspondingly \( a_{r1} = \cos \theta = [-1, 0, 1] \)) and modulo weight \( w_r = 1 \) which are shown in Figure 2 a and e. For inversion pattern, we have \( W^r_{r2} = \tilde{W}^r_{r3} \) (correspondingly \( a_{r2} - a_{r3} = 0 \) and \( b_{r2}c_{r3}d_{r2} + b_{r3}c_{r2}d_{r3} = 0 \)), which are shown in Figure 2 b,c,d and f,g,h. For composition pattern, we have \( W^r_{r4} = W^r_{r6} \otimes W^r_{r5} \). We show the real part and the first imaginary part (correspondingly \( a_{r4} - (a_{r6}a_{r5} - b_{r6}b_{r5} - c_{r6}c_{r5} - d_{r6}d_{r5}) = 0 \) and \( b_{r4} - (a_{r6}b_{r5} + a_{r5}b_{r6} + c_{r6}d_{r5} - c_{r5}d_{r6}) = 0 \)) in Figure 2 i, j, k and l. Table 7 summarizes the MRR for each relation on WN18RR, confirming the superior representation capability of HRQE in modelling different types of relation.

**Analysis on Hierarchy-Aware.** We plot the entity embeddings of two models: RQE and HRQE. Their entities are all pure quaternions. For an intuitive display, we project it to a 2D plane and display them in polar coordinates. The radius \( r \) of the polar coordinates is quaternion norm \( |Q| \), and the angle is twice the angle between the entities and \( i + j + k \). Note that we use the logarithmic scale to better display the differences between entity embeddings. As all the moduli have values less than one, after applying the logarithm operation, the larger radii in the figures will actually represent smaller modulus. Compared with the tail entities, the head entities in Figure 3 a, b, and c are at lower levels, similar levels, and higher levels in the semantic hierarchy, respectively. We can see that there exist clear hierarchies in HRQE, which demonstrates that the modulus part in HRQE can help effectively model the semantic hierarchies.

**Analysis on Complex Relations.** We also conduct further investigation on the performance of HRQE on complex relations: 1-1-N, N-1-1, and N-1-N relations. We compare with QuatE and QuatRE (Nguyen et al., 2022). QuatRE adds two additional relational quaternions and quaternion mul-

![Figure 3: Visualization of the embeddings of several entity pairs from WN18RR dataset.](image)

Table 7: MRR for the models tested on each relation of WN18RR.
tuples with the head and tail entities to improve QuatE’s ability for handling complex relations. Table 6 shows the MRR and H@10 scores for predicting the head entities and then the tail entities with respect to each relation category on FB15k-237, wherein our HRQE outperforms QuatE and QuatRE on these relation categories.

6 Conclusion

To model various relation patterns and semantic hierarchies in knowledge graphs, we propose a novel knowledge graph embedding model HRQE, which maps entities into 3D space with rotation and modulo constraints. Empirical experimental evaluations on benchmark datasets show that our proposed HRQE significantly outperforms several existing state-of-the-art methods. Further investigation shows that HRQE is capable of modeling relations with various relation patterns and modeling entities at both different levels and the same levels in the semantic hierarchies.

Acknowledgement

This work was supported in part by National Key R&D Program of China Grand No. 2020YFB1708002, and NNSFC Grant No. 61971008.

References

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In Proc. of Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, pages 5185–5194.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In Proc. of ACM International Conference on Management of Data, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Proc. of Annual Conference on Neural Information Processing Systems, pages 1–9.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochn Cao, and Qingming Huang. 2021. Dual quaternion knowledge graph embeddings. In Proc. of AAAI Conference on Artificial Intelligence, volume 35, pages 6894–6902.

Ines Chami, Adv Wolf, Da-Cheng Juan, Federic Sala, Sujith Ravi, and Christopher Re. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In Proc. of Annual Meeting of the Association for Computational Linguistics, pages 6901–6914.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In Proc. of AAAI Conference on Artificial Intelligence, volume 32, pages 1811–1818.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In Proc. of AAAI Conference on Artificial Intelligence.

Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In Proc. of ACM International Conference on Information & Knowledge Management, pages 385–394.

William Rowan Hamilton. 1844. Lxxviii. on quaternions; or on a new system of imaginaries in algebra: To the editors of the philosophical magazine and journal. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 25(169):489–495.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In Proc. of AAAI Conference on Artificial Intelligence, pages 985–991.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. Proc. of Annual Conference on Neural Information Processing Systems, 31.

Denis Krompaß, Stephan Baier, and Volker Tresp. 2015. Type-constrained representation learning in knowledge graphs. In Proc. of International semantic web conference, pages 640–655.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In Proc. of International Conference on Machine Learning, pages 2863–2872.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Klee, Sören Auer, et al. 2015. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web, 6(2):167–195.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In Proc.
Appendix

A.1 Definitions of Different Relation Patterns

Definition 1 Relations $r_i$ are multiple if $\forall i \in 0, ..., m$, $(h, r, t)$ can hold in knowledge graphs simultaneously. A clause with such form is a multiple relations pattern.

Definition 2 A relation $r$ is symmetric (antisymmetry) if $\forall x, y$

$$r(x, y) \Rightarrow r(y, x) \ (r(x, y) \Rightarrow \neg r(y, x)).$$

A clause with such form is a symmetry (antisymmetry) pattern.

Definition 3 Relation $r_1$ is inverse to relation $r_2$ if $\forall x, y$

$$r_2(x, y) \Rightarrow r_1(y, x).$$

A clause with such form is an inversion pattern.

Definition 4 Relation $r_1$ is composed of relation $r_2$ and relation $r_3$ if $\forall x, y, z$

$$r_2(x, y) \land r_3(y, z) \Rightarrow r_1(x, z).$$

A clause with such form is a composition pattern.

Definition 5 For each relation $r$, we compute average number of tails per head (tphr) and average number of heads per tail (hptr). If $tphr < 1.5$ and $hptr < 1.5$, $r$ is treated as $1$-to-$1$; if $tphr > 1.5$ and $hptr > 1.5$, $r$ is treated as $N$-to-$N$; if $tphr > 1.5$ and $hptr < 1.5$, $r$ is treated as $1$-to-$N$. Clauses with such form are complex relations.
A.2 Proof of Lemma 1

Proof of symmetry pattern. When $\theta = [0^\circ, 180^\circ, 270^\circ, 360^\circ]$, HRQE can represent a symmetric relationship, then $W_r^\theta = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} (q_i + u_j + v_k)$, we need to prove:

$$W_r^\theta \otimes Q_h \otimes W_r^\theta \cdot Q_t = W_r^\theta \otimes Q_t \otimes W_r^\theta \cdot Q_h$$

(9)

For $\theta = [0^\circ, 360^\circ]$, we have $W_r^\theta = \pm 1$. Firstly, we expand the left term:

$$W_r^\theta \otimes Q_h \otimes \tilde{W}_r^\theta \cdot Q_t = Q_h \cdot \tilde{Q}_t = (x_h, x_t) + (y_h, y_t) + (z_h, z_t).$$

(10)

We then expand the right term:

$$W_r^\theta \otimes Q_t \otimes \tilde{W}_r^\theta \cdot Q_h = \tilde{Q}_t \cdot Q_h = (x_h, x_t) + (y_h, y_t) + (z_h, z_t).$$

(11)

We can easily see that those two terms are equal.

Proof of antisymmetry pattern. In order to prove the antisymmetry pattern, we need to prove the following inequality when $\theta \neq [0^\circ, 180^\circ, 270^\circ, 360^\circ]$:

$$W_r^\theta \otimes Q_h \otimes \tilde{W}_r^\theta \cdot Q_t \neq W_r^\theta \otimes Q_t \otimes \tilde{W}_r^\theta \cdot Q_h$$

(14)

Firstly, we expand the left term:

$$W_r^\theta \otimes Q_h \otimes \tilde{W}_r^\theta \cdot Q_t = [((qq - uu - vv)x_h + 2(qu)y_h + 2(qv)z_h)i + (2(qu)x_h + (qq + uu - vv)y_h + 2(uv)z_h)j + (2(qv)x_h + 2(uv)y_h + (qq - uu + vv)z_h)k] \cdot (x_h i + y_h j + z_h k)

+ (x_h, (qq - uu - vv), x_t) + (x_h, 2(qu), y_t)
+ (x_h, 2(qv), z_t) + (y_h, 2(qu), x_t)
+ (y_h, (qq + uu - vv), y_t) + (y_h, 2(uv), z_t)
+ (z_h, 2(qv), x_t) + (z_h, 2(uv), y_t)
+ (z_h, (qq - uu + vv), z_t).$$

(12)

We then expand the right term:

$$W_r^\theta \otimes Q_t \otimes \tilde{W}_r^\theta \cdot Q_h = [((pp + qq - uu - vv)x_h + 2(-pu + vq)y_h + 2(pq + uv)z_h)i + (pp + qq - uu - vv)x_h] + 2(pu + vq)z_h)j + (2(-pu + vq)x_h + 2(pq + uv)y_h + (pp - qq - uu + vv)z_h)k] \cdot (x_h i + y_h j + z_h k)

+ (x_h, (pp + qq - uu - vv), x_t) + (x_h, 2(pv + qu), y_t)
+ (x_h, 2(-pu + vq), z_t) + (y_h, 2(-pv + qu), x_t)
+ (y_h, (pp - qq - uu - vv), y_t)
+ (y_h, 2(pq + uv), z_t) + (z_h, 2(pu + vq), x_t)
+ (z_h, 2(-pq + uv), y_t)
+ (z_h, (pp - qq - uu + vv), z_t).$$

(15)
We expand the right term:

\[
\begin{align*}
W_r^a \odot Q_i \odot \bar{W}_r^a \cdot Q_h &= \left(\{(pp + qq - uu - vv)x_t, 2(-pv + qu)y_t, + 2(pu + qv)z_t\}i + (2pv + qu)x_t + (pp - qq + uu - vv)y_t + 2(-pq + uv)z_t, 2(pq + uv)y_t + (pp - qq - uu + vv)z_t\}\right)k \\
&= \left(\langle x_t, (pp + qq - uu - vv), x_h \rangle + \langle y_t, 2(pv + qu), y_h \rangle + \langle y_t, 2(-pv + qu), x_h \rangle + \langle y_t, (pp - qq + uu - vv), y_h \rangle + \langle y_t, 2(pq + uv), z_h \rangle + \langle z_t, 2(pq + uv), y_h \rangle + \langle z_t, (pp - qq - uu + vv), z_h \rangle \right).
\end{align*}
\]

(16)

We can easily see that those two terms are not equal as the signs for some terms are not the same.

A.3 Proof of Lemma 2

Proof of inversion pattern. To prove the inversion pattern, we need to prove that:

\[
W_r^a \odot Q_h \odot \bar{W}_r^a \cdot Q_t = \bar{W}_r^a \odot Q_t \odot W_r^a \cdot Q_h
\]

(17)

We expand the right term:

\[
\begin{align*}
\bar{W}_r^a \odot Q_t \odot W_r^a \cdot Q_h &= \left(\langle x_t, (pp + qq - uu - vv), x_h \rangle + \langle y_t, 2(pv + qu), y_h \rangle + \langle y_t, 2(-pv + qu), x_h \rangle + \langle y_t, (pp - qq + uu - vv), y_h \rangle + \langle y_t, 2(pq + uv), z_h \rangle + \langle z_t, 2(pq + uv), y_h \rangle + \langle z_t, (pp - qq - uu + vv), z_h \rangle \right)k \\
&= \langle x_t, (pp + qq - uu - vv), x_h \rangle + \langle y_t, 2(pv + qu), y_h \rangle + \langle y_t, 2(-pv + qu), x_h \rangle + \langle y_t, (pp - qq + uu - vv), y_h \rangle + \langle y_t, 2(pq + uv), z_h \rangle + \langle z_t, 2(pq + uv), y_h \rangle + \langle z_t, (pp - qq - uu + vv), z_h \rangle.
\end{align*}
\]

(18)

We can easily check the equality of these two terms.

A.4 Proof of Lemma 3

Proof of composition relation. For composition relations we can get that:

\[
W_{r3}^a \odot (W_{r2}^a \odot Q_h \odot \bar{W}_{r2}^a) \odot \bar{W}_{r1}^a \cdot Q_t
\]

\[
= (W_{r3}^a \odot W_{r2}^a) \odot Q_h \odot (W_{r2}^a \odot W_{r1}^a) \cdot Q_t
\]

(19)

A.5 Summary of Several Popular Knowledge Graph Embedding Models

Table 8 summarizes several popular knowledge graph embedding models, including scoring func-
tions, parameters, and time complexities. TransE, HolE, and DistMult use Euclidean embeddings, while ComplEx and RotatE operate in the complex space. QuatE, DualE (dual quaternion) and our models operate in the quaternion space. HAKE and our model HRQE can learn hierarchy-aware in knowledge graphs.

A.6 Parameter Settings

We list the best hyperparameter settings of RQE and HRQE w.r.t. the validation dataset on several benchmarks in Table 9 and Table 10.

| Dataset | \( n_B \) | \( k \) | \( \lambda_1 \) | \( \lambda_2 \) | \( n_{\text{neg}} \) |
|--------|--------|--------|--------|--------|--------|
| WN18   | 10     | 300    | 0.03   | 0.0    | 10     |
| FB15K  | 100    | 400    | 0.05   | 0.0    | 10     |
| WN18RR | 10     | 300    | 0.3    | 0.3    | 2      |
| FB15K237 | 100 | 500    | 0.5    | 0.01   | 10     |

Table 9: Hyperparameters for RQE

| Dataset | \( n_B \) | \( k \) | \( \lambda_1 \) | \( \lambda_2 \) | \( n_{\text{neg}} \) |
|--------|--------|--------|--------|--------|--------|
| WN18   | 10     | 300    | 0.05   | 0.01   | 10     |
| FB15K  | 100    | 500    | 0.03   | 0.0    | 10     |
| WN18RR | 10     | 300    | 0.3    | 0.01   | 1      |
| FB15K237 | 100 | 500    | 0.5    | 0.01   | 10     |

Table 10: Hyperparameters for HRQE