

Dual Quaternion Knowledge Graph Embeddings

Zongsheng Cao^{1,2}, Qianqian Xu^{3,*}, Zhiyong Yang^{1,2}, Xiaochun Cao^{1,2,6}, Qingming Huang^{3,4,5,6,*}

¹ State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China

² School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

³ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, China

⁴ School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China

⁵ Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, China

⁶ Peng Cheng Laboratory, Shenzhen, China

{caozongsheng,yangzhiyong,caoxiaochun}@iie.ac.cn, xuqianqian@ict.ac.cn, qmhuang@ucas.ac.cn

Abstract

In this paper, we study the problem of learning representations of entities and relations in the knowledge graph for the link prediction task. Our idea is based on the observation that the vast majority of the related work only models the relation as a single geometric operation such as translation or rotation, which limits the representation power of the underlying models and makes it harder to match the complicated relations existed in real-world datasets. To embrace a richer set of relational information, we propose a new method called dual quaternion knowledge graph embeddings (**DualE**), which introduces dual quaternions into knowledge graph embeddings. Specifically, a dual quaternion behaves like a “complex quaternion” with its real and imaginary part all being quaternary. The core of DualE lies a specific design of dual-quaternion-based multiplication, which universally models relations as the compositions of a series of translation and rotation operations. The major merits of DualE are three-fold: 1) it is the first unified framework embracing both rotation-based and translation-based models in 3D space, 2) it expands the embedding space to the dual quaternion space with a more intuitive physical and geometric interpretation, 3) it satisfies the key patterns and the multiple relations pattern of relational representation learning. Experimental results on four real-world datasets demonstrate the effectiveness of our DualE method.

Introduction

Knowledge Graph (KG) represents a collection of interlinked descriptions of entities, namely, real-world objects, events, situations, or abstract concepts. During the past decade, KG has been proven to be an indispensable build-block for a wide spectrum of applications ranging from question answering, knowledge inference to natural language processing. To effectively integrate KG into downstream AI applications, a key step, known as Knowledge Graph Embedding (KGE), then comes into play as a powerful tool which encodes entities and relations of the graph into low-dimensional representations.

We could roughly divide the vast majority of KGE methods into two general families based on how relations in KG are formulated. As the name suggests, translation family refers to the models which regard relations as translations, which

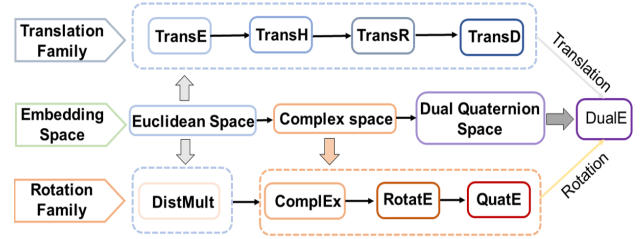


Figure 1: An illustration of that DualE is a unified framework of translation family and rotation family.

could be traced back to TransE (Bordes et al. 2013). A remarkable trait of the models belonging to this family is that they provide a natural way to represent the hierarchical relations which are extremely common in KGs. And its variants such as (Wang et al. 2014; Lin et al. 2015) can model the multiple relations pattern. Meanwhile, rotation family refers to the models in which the relations are modeled as rotation operations, which typically includes (Sun et al. 2019; Zhang et al. 2019). As what is claimed in the representative work RotatE (Sun et al. 2019), rotation can model all the three fundamental patterns of relations in KG, *i.e.*, symmetry/antisymmetry, inversion, and composition. A thorough discussion of these methods is delayed to the next section.

With the efforts of the translation family and rotation family models, we have witnessed great success of KG-based applications. Nonetheless, a single translation or rotation is not always a better way to represent relations. For example, translation family cannot model all the three fundamental patterns (Sun et al. 2019); rotation family models have little effect on hierarchical relations and multiple relations pattern. See the motivation part for more detailed analysis on this issue. As such, we find that the strengths of translation family and rotation family are essentially complementary. This brings about the question that is there a way to unify both translation and rotation in one framework?

Fortunately, we can seek out a solution by means of a number field named dual quaternion, which is proposed by William Kingdom Clifford (Clifford 1871). The dual quaternion has an intuitive geometric and physical interpretation: (1) It can represent both translation and rotation. (2) It pro-

vides an elegant way of solving a range of problems that are otherwise complex, such as rigid transformations.

In this paper, we propose a new method called dual quaternion knowledge graph embeddings (DualE). As a key instrument, we introduce the dual quaternion as the number field of the embedding space. Specifically, embeddings in a dual quaternion space are vectors in the hypercomplex space \mathbb{H}_d formed as $a + \epsilon b$, where a and b are two quaternions representing the real and dual part of the vector, respectively. It could be proved that with a proper definition of dual-quaternion-based multiplication, we can represent relations in the KGs as compositions of translation and rotation operations. As shown in Figure 1, this allows us to unify the previous studies in the translation family and rotation family.

To summarize, our contributions are as follows: 1) We provide theoretical analyses on why both rotation and translation operations are necessary for knowledge graph embedding. 2) We propose a new framework called DualE to effectively integrate rotation and translation operations in KG embedding methods. 3) We conduct a series of theoretical and empirical analyses to show the strength of DualE against some of the SOTA methods.

Related Work

Recall that we divide the majority of KGE methods roughly into two families according to the way they manipulate relations. In this section, we take a further step to provide a detailed review of the methods in each family.

Translation Family. This family includes TransE (Bordes et al. 2013) and its variants. The common trait of these methods is that they all model relations as translations between heads and tails in the embedding space. TransE (Bordes et al. 2013) is the first model that realizes this assumption based on the principle $head + relation \approx tail$. TransH (Wang et al. 2014), TransR (Lin et al. 2015), TransD (Ji et al. 2015), and TransA (Xiao et al. 2015) then improve this idea with different projection strategies. TransG (Xiao, Huang, and Zhu 2016) and KG2E (He et al. 2015) further inject probabilistic principles into this framework by considering Bayesian nonparametric Gaussian mixture model and Gaussian distribution covariance, respectively. TransSparse (Ji et al. 2016) provides adaptive sparsity to the transfer matrices in search of a solution to heterogeneity and imbalance issues of KGs. Last but definitely not the least, a recent work called TorusE (Ebisu and Ichise 2018) adopts torus to avoid forcing embedding to be on a sphere. In one word, translation family models provide a simple yet effective way to achieve better results than previous complex models such as (Bordes et al. 2014, 2011; Sutskever, Tenenbaum, and Salakhutdinov 2009). But unfortunately, translation family cannot completely capture all the three fundamental patterns of relations in KG, *i.e.*, symmetry/antisymmetry, inversion, and composition.

Rotation Family. Another branch of studies, what we call rotation family here, emerges as an alternative way to learn embedding in a complex space. This spirit is initiated by DistMult (Yang et al. 2015) and ComplEx (Trouillon et al. 2016), which extends the embedding space to the complex space for the first time. RotatE (Sun et al. 2019) then proposes to formulate relations as rotations from the head entity to the

tail entity through rotational operation in the complex space with only one rotating surface. As a remarkable property, RotatE is the first model to unify symmetry/antisymmetry, inversion, and composition patterns for KGE. This suggests that rotation operations in the complex space have a strong potential to empower a universal knowledge representation. This is why we name this direction after rotation. Most recently, QuatE (Zhang et al. 2019) extends the complex space into the quaternion space with two rotating surfaces. However, rotation family cannot model the hierarchical structure, nor can it model multiple relations between two entities simultaneously.

KGE with Deep Learning. Recently there are some models using neural networks to produce KG embeddings with remarkable effects. For instance, R-GCN (Schlichtkrull et al. 2018), ConvE (Dettmers et al. 2017) and ConvKB (Nguyen et al. 2018), KBGAT (Nathani et al. 2019), A2N (Bansal et al. 2019). A downside of these methods is that the geometric meaning is not clear or the transformation is single.

Problem Setup

Problem Definition

In the KGE problem, we define the head entity, the relation and the tail entity as h , r and t , respectively. Now we define a triplet as (h, r, t) , which represents $h \xrightarrow{r} t$, *i.e.*, the head entity h is related to tail entity t by a relation r . Then a knowledge graph could be represented by means of a set of specific triplets $(h, r, t) \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, where \mathcal{V} and \mathcal{R} are entity and relation sets, respectively. Practically, it is impossible to collect a complete knowledge graph with pure human efforts. The link prediction task then comes into play as a powerful application to automatically infer missing links in a knowledge graph. From a practical perspective, our ultimate goal is then to develop effective knowledge graph embedding algorithms for the link prediction task. In this way, our algorithm will simultaneously produce proper embedding of the nodes/relations as well as the confidence score of a given triplet (h, r, t) .

Motivation

Before introducing the motivation, we give definitions of the key patterns (see Appendix for the definitions of symmetry/antisymmetry, inversion, composition patterns) and multiple relations pattern first.

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

Recalling what we have aforementioned in the introduction, our idea is inspired by the fact that the single effect of either translation or rotation might be insufficient to capture the realistic structure of relations. To show this, we consider a knowledge graph as follows. Hunk is the father and guardian of John and Bruce is the father of Sam, then we have relations: John $\xrightarrow{\text{father}}$ Hunk and Sam $\xrightarrow{\text{father}}$ Bruce; Bruce $\xrightarrow{\text{guardian}}$ Hunk and Hunk are neighbors, then we have the relation: Bruce $\xrightarrow{\text{neighbor}}$ Hunk; John and Sam are classmates, we have the

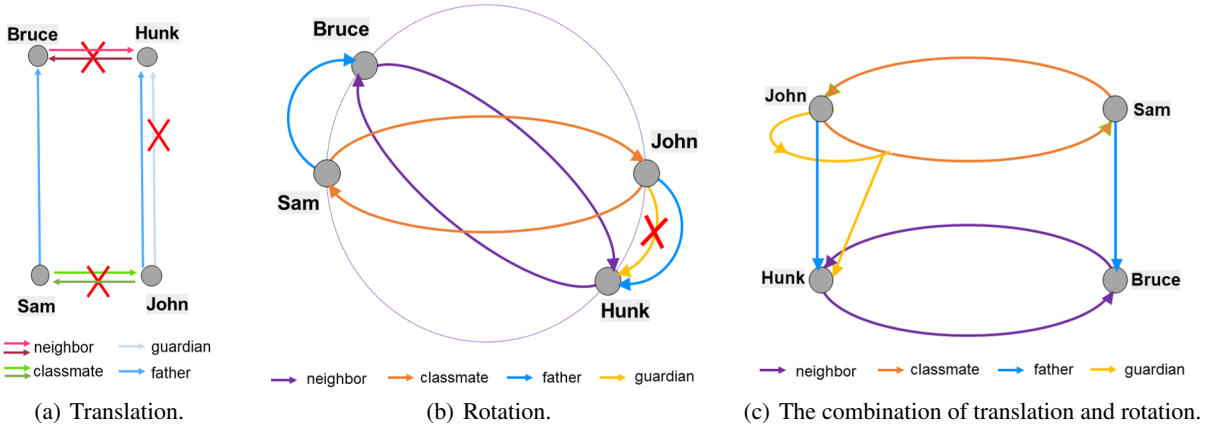


Figure 2: Illustrations of different transformations modeling relations. Note that the arc represents the operation of rotation, not the trajectory of rotation. We can see that combining translation and rotation will diversify the description of the relation.

relation: $\text{John} \xleftrightarrow{\text{classmate}} \text{Sam}$. As shown in Figure 2(a), we can see that translation method can model the hierarchical structure of relations, but cannot model symmetric relations. As shown in Figure 2(b), we can see that rotation method can model symmetric relations, but cannot model the hierarchical structure and multiple relations. In this sense, it becomes crucial to unify both translation and rotation. In fact, as shown in Figure 2(c), we can see that the combination of rotation and translation can overcome their respective shortcomings well.

Next, we consider a more complicated example where both symmetry and inversion are necessary. For example, for Hunk and John, we first have the relation: $\text{Hunk} \xrightarrow{\text{son}} \text{John}$; if they graduated from the same college, we have the relation: $\text{Hunk} \xleftrightarrow{\text{alumni}} \text{John}$. Because all the translation family cannot model both symmetry and inversion relations, it is invalid for this pattern. In the embedding, the angle between Hunk and John is fixed, and a relation can only correspond to a fixed angle of rotation. So an angle cannot model these two relations at the same time, otherwise it will cause confusion. Therefore, rotation cannot model this pattern. For the combination of translation and rotation, we can see that modeling the symmetry relation as rotation and modeling the inversion relation as translation can solve the problem easily.

Let us further consider a more general question: if there are many multiple relations between the two entities, can the combination of rotation and translation model these relations simultaneously? Obviously, first of all, a single translation or rotation is invalid for this multiple relations pattern. Then some translation family models such as TransR can solve this problem. For each type of relation in TransR, it has not only a vector r to model itself but also a mapping matrix M_r to model the relation space while they cannot model inversion and composition patterns. Contrary to such complicated methods, we model the multiple relations pattern by using the combination of translation and rotation while it can model all the key patterns. As shown in Table 1, we show the pattern modeling and inference abilities of several transformations.

Pattern	Translation	Rotation	Translation+Rotation
Symmetry	×	✓	✓
Antisymmetry	✓	✓	✓
Inversion	✓	✓	✓
Composition	✓	✓	✓
Multiple	✓	×	✓

Table 1: The pattern modeling and inference abilities of several transformations (See Appendix for more details).

We can see that it strongly supports our motivation for the combination of rotation and translation.

Basic Properties of Dual Quaternion Numbers

In this section, we will introduce dual quaternions by quaternions and dual numbers, along with its operations and properties.

Quaternion: Quaternion is a hypercomplex number system firstly described by Hamilton (Hamilton 1844). A quaternion q is defined to be the sum of a scalar q_0 and a vector $\mathbf{q} = (q_1, q_2, q_3)$ namely, $q = q_0 + \mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the unit vectors along the x -, y -, z -axes, respectively. The quaternion can also be viewed as a 4-tuple (q_0, q_1, q_2, q_3) . A vector \mathbf{v} is called a pure quaternion in the form of $(0, q_1, q_2, q_3)$.

Dual numbers: A dual number is defined to be $z_d = a + \epsilon a_\epsilon$ where a and a_ϵ are real numbers, or more generally, elements of a (algebraic) field, and ϵ is a dual unit with $\epsilon^2 = 0$. In the formula above, a and a_ϵ are referred to as the **real** and **dual** parts of z_d .

Dual quaternion: Dual quaternion algebra is an extension of the dual-number theory by Clifford (Clifford 1871) in an effort to combine with Hamilton’s quaternion algebra.

A dual quaternion Q has the form $Q = a + \epsilon b$, where a and b are quaternions given by below:

$$a = a_0 + a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}, \quad b = b_0 + b_1\mathbf{i} + b_2\mathbf{j} + b_3\mathbf{k}.$$

The quaternions a and b are the real and dual part of Q , respectively. We can also view Q as an 8-tuple:

$$Q = (a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3).$$

Please see Appendix¹ for operational properties of dual quaternion.

Dual quaternion can model both translation and rotation.

Let $q = \cos \frac{\theta}{2} + \hat{u} \sin \frac{\theta}{2}$ be a quaternion that represents a rotation about the unit vector \hat{u} through θ . We can have $|q| = 1$. Then we denote the corresponding rotation matrix as R and let $t = (t_1, t_2, t_3)$ be a translation, which can be set as a pure quaternion. A point v under the rotation R followed by the translation t becomes the point $Rv + t$, then the translation vector sequence R, t can be compactly represented by a dual quaternion σ , and it can be written as

$$\sigma = q + \frac{\epsilon}{2} tq. \quad (1)$$

Particularly, if the transformation is a pure rotation, i.e., $t = 0$, we end up with $\sigma = q$. If the transformation is a pure translation, that is, $\theta = 0$, we end up with $\sigma = 1 + \frac{\epsilon}{2} t$. Note that Eq.(1) is a unit dual quaternion, and please see Eq.(15) in Appendix for the proof and more details.

Methodology

Dual Quaternion Representations for Knowledge Graph Embedding

In this section, we propose an effective knowledge graph embedding framework named as DualE. First of all, we elaborate the details of the framework. After that, we provide a series of analyses to show the strength of our framework.

Symbol Description. Suppose that we have a knowledge graph \mathcal{G} consisting of N entities and M relations. We formulate the all entity embeddings as a dual quaternion matrix $Q \in \mathbb{H}_d^{N \times k}$, where each row is an embedding vector for a specific entity of dimensionality k , and denote the relation embeddings as $W \in \mathbb{H}_d^{M \times k}$. Given a triplet (h, r, t) , the embedding of head entity h is denoted as $Q_h = (a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3)$ and the embedding of the tail entity $Q_t = (e_0, e_1, e_2, e_3, f_0, f_1, f_2, f_3)$, where $Q_h, Q_t \in Q$. Then we denote the relation r as $W_r = (c_0, c_1, c_2, c_3, d_0, d_1, d_2, d_3)$, where $W_r \in W$.

Normalization of the relation dual quaternion. As mentioned in (Clifford 1871), the unit dual quaternions can represent translation and rotation. In order to avoid the impact of scaling, we need to use some steps of Schmidt orthogonalization to normalize the relation dual quaternion W_r to a unit relation dual quaternion W_r^\diamond . We define $W_r = (c, d)$, where $c = (c_0, c_1, c_2, c_3)$, $d = (d_0, d_1, d_2, d_3)$ and we denote

$$\bar{d} = d - \frac{(d, c)}{(c, c)} c = (\bar{d}_0, \bar{d}_1, \bar{d}_2, \bar{d}_3). \quad (2)$$

We then obtain a normalized variable called c' :

$$c' = \frac{c}{\|c\|} = \frac{c_0 + c_1 i + c_2 j + c_3 k}{\sqrt{c_0^2 + c_1^2 + c_2^2 + c_3^2}}. \quad (3)$$

¹For the appendix and the code, please refer to <https://github.com/Lion-ZS/DualE>.

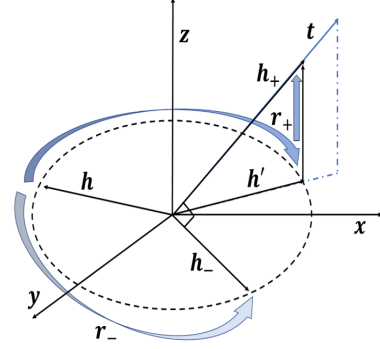


Figure 3: DualE models relation r as translation and rotation in 3D space.

Then we denote $W_r^\diamond = (c', \bar{d}) = (c, d) = (c_0, c_1, c_2, c_3, d_0, d_1, d_2, d_3)$. And we can deduce that:

$$c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1. \quad (4)$$

$$c_0 d_0 + c_1 d_1 + c_2 d_2 + c_3 d_3 = 0. \quad (5)$$

It can be seen that Eq.(4)-(5) make the number of degrees of freedom reduce from 8 to 6, whose physical interpretation happens to be the degrees of freedom of the rigid body in a 3D world.

Translate and rotate the head entity. As shown in Figure 3, for DualE, if a triplet exists in the KG, we will rotate h to the place of h' and translate h' to the place of h_+ to make the angle between h_2 and tail t to be zero (shown as r_+). Otherwise, we can make the head and tail entity be orthogonal so that their product becomes zero (shown as r_-). Then we define an intermediate variable Q'_h as the result of the multiplication between Q_h and W_r^\diamond :

$$\begin{aligned} Q_h \otimes W_r^\diamond &= (a_h \odot p - b_h \odot q - c_h \odot u - d_h \odot v) \\ &\quad + (a_h \odot q + b_h \odot p + c_h \odot v - d_h \odot u) i \\ &\quad + (a_h \odot u - b_h \odot v + c_h \odot p + d_h \odot q) j \\ &\quad + (a_h \odot v + b_h \odot u - c_h \odot q + d_h \odot p) k, \end{aligned} \quad (6)$$

where \odot denotes the element-wise multiplication between two vectors. Expand Eq.(6), and we can get the following form of Q'_h , which represents the head entity after translation and rotation:

$$\begin{aligned} Q'_h &= (h_0 + \epsilon h'_0) + (h_1 + \epsilon h'_1) i + (h_2 + \epsilon h'_2) j \\ &\quad + (h_3 + \epsilon h'_3) k \\ &= a'_h + b'_h i + c'_h j + d'_h k, \end{aligned} \quad (7)$$

where $h_0, h'_0, h_1, h'_1, h_2, h'_2, h_3, h'_3$ are obtained by merging related items after the calculation.

Scoring function and Loss function: Then the scoring function $\phi(h, r, t) = \langle Q'_h, Q_t \rangle$ is defined by the dual quaternion inner product naturally:

$$\phi(h, r, t) = \langle a'_h, a_t \rangle + \langle b'_h, b_t \rangle + \langle c'_h, c_t \rangle + \langle d'_h, d_t \rangle. \quad (8)$$

Then we adopt the cross entropy loss as our loss function. And we use Ω and $\Omega' = \mathcal{E} \times \mathcal{R} \times \mathcal{E} - \Omega$ to denote the set of observed triplets and the set of unobserved triplets, respectively. Moreover, we perform regularization on \mathbf{Q} and \mathbf{W} to avoid overfitting, where we learn parameters to regularize \mathbf{Q} and \mathbf{W} by using ℓ_2 norm with regularization rates λ_1 and λ_2 :

$$L(\mathbf{Q}, \mathbf{W}) = \sum_{r(h,t) \in \Omega \cup \Omega^-} \log(1 + \exp(-Y_{hrt} \phi(h, r, t))) + \lambda_1 \|\mathbf{Q}\|_2^2 + \lambda_2 \|\mathbf{W}\|_2^2, \quad (9)$$

where $\Omega^- \subset \Omega$, $Y_{hrt} \in \{-1, 1\}$ represents the corresponding label of the triplet (h, r, t) . We use negative sampling strategies which include sampling, adversarial sampling (Sun et al. 2019), and Bernoulli sampling (Wang et al. 2014) to sample Ω^- from the unobserved set Ω' . Under the premise that the search space can be effectively limited, we optimize the loss function by utilizing Adagrad (Duchi, Hazan, and Singer 2011). Please see Appendix in Algorithm 1 for more details of the training algorithm and the initialization scheme.

Theoretical Analysis

In this part, we discuss the theoretical properties of DualE. As shown in Table 5 in Appendix, we provide comparison between our proposed method and several popular models in terms of the scoring function and the time complexity.

Dual quaternion space is non-commutative and associative. The algebraic properties of the relations depend on the corresponding properties of the multiplication. As mentioned in QuatE, the composition is realized through the lens of the Hamilton product (multiplication of quaternions) between relations. In the hypercomplex space, we also notice that there is an octonion model². Similar to dual quaternion, the octonions also live in an 8D world. In the forthcoming discussion, we will reveal the fundamental advantage of dual quaternions, which makes it a better choice. Regarding multiplication, the dual quaternion is non-commutative and associative, while the octonion is non-commutative and non-associative. These properties have a great impact on the embedding of relations.

Suppose that there are three persons “ x, y, z ” and the relations between the three are as follows: $x \xrightarrow[r_1]{\text{father}} y \xrightarrow[r_2]{\text{mother}} z$. The relations above mean that y is the father (denoted as r_1) of x and z is the mother (denoted as r_2) of y , we can easily infer that z is the father’s mother ($r_1 \circ r_2$, the \circ here represents composition) of x , i.e., z is the paternal grandmother of x . But we cannot infer that z is the mother’s father ($r_2 \circ r_1$) of x , which represents that z is the maternal grandfather of x . So we can get that the non-commutative of the multiplication is necessary.

Let us consider another example, and suppose there are four persons “ w, x, y, z ” and the relations between the four are as follows: $w \xrightarrow[r_1]{\text{father}} x \xrightarrow[r_2]{\text{father}} y \xrightarrow[r_3]{\text{mother}} z$. From relations above, we can easily infer that z is the father’s father’s

mother ($r_1 \circ r_2 \circ r_3$) of w . Then we can say that z is the paternal grandfather’s mother ($(r_1 \circ r_2) \circ r_3$) of w , and we can also say that z is the father’s paternal grandmother ($r_1 \circ (r_2 \circ r_3)$) of w , for the relations they express are the same. So we can get that the associative of the multiplication is appropriate. The weakness of octonions is also verified by the experiments in (Zhang et al. 2019).

Capability in Modeling Symmetry, Antisymmetry, Inversion and Composition. As mentioned by (Sun et al. 2019), the three key types of relation patterns above are very important and widely spread in knowledge graphs. So can DualE portray these three key relation patterns and multiple relations pattern? The answer is positive in light of the following lemmas.

Lemma 1 *DualE can infer the (anti)symmetry pattern. (See Appendix 1 for the proof)*

Lemma 2 *DualE can infer the inversion pattern. (See Appendix 2 for the proof)*

Lemma 3 *DualE can infer the composition pattern. (See Appendix 3 for the proof)*

Lemma 4 *DualE can infer the multiple relations pattern. (See Appendix 4 for the proof)*

Connection to rotation family and translation family models. The following results show that DualE is a unified framework for rotation family and translation family. And then, we have the following lemmas:

Lemma 5 *DualE can degenerate into rotation family models. (See Appendix 5 for the proof)*

Lemma 6 *DualE can degenerate into translation family models. (See Appendix 6 for the proof)*

Lemma 7 *Rotation family models cannot derive DualE model. (See Appendix 10 for the proof)*

Lemma 8 *Translation family models cannot derive DualE model. (See Appendix 11 for the proof)*

And based on the lemmas above, we can derive the following theorem:

Theorem 1 *DualE is a unified framework embracing rotation family and translation family.*

Proof Combining Lemma 5, Lemma 6, Lemma 7, Lemma 8, we can obviously prove the theorem.

Experiments and Results

Experimental Setup

In this section, we conduct a series of empirical studies on four widely-used knowledge graph datasets, which are summarized in Table 3.

Datasets: WN18 (Bordes et al. 2013) is a subset of WordNet (Miller 1995), a database featuring lexical relations between words. The dataset also has many inverse relations. So mainly the relation patterns in WN18 are **symmetric/antisymmetric** and **inversion**. FB15K (Bordes et al. 2013) is a subset of Freebase (Bollacker et al. 2008) and is a large-scale knowledge graph containing general knowledge facts. Therefore, the key of link prediction on FB15K

²Hypercomplex number with one real part and seven imaginary parts: $O = x_0 + x_1 e_1 + x_2 e_2 + x_3 e_3 + x_4 e_4 + x_5 e_5 + x_6 e_6 + x_7 e_7$

Model	WN18					FB15K				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	-	0.495	0.943	0.888	0.113	-	0.463	0.749	0.578	0.297
DistMult	655	0.797	0.946	-	-	42.2	0.798	0.893	-	-
HolE	-	0.938	0.949	0.945	0.930	-	0.524	0.739	0.759	0.599
ComplEx	-	0.941	0.947	0.945	0.936	-	0.692	0.840	0.759	0.599
ConvE	374	0.943	0.956	0.946	0.935	51	0.657	0.831	0.723	0.558
R-GCN+	-	0.819	0.964	0.929	0.697	-	0.696	0.842	0.760	0.601
Simple	-	0.942	0.947	0.944	0.939	-	0.727	0.838	0.773	0.660
NKGE	336	0.947	0.957	0.949	0.942	56	0.73	0.871	0.790	0.650
TorusE	-	0.947	0.954	0.950	0.943	-	0.733	0.832	0.771	0.674
RotatE	184	0.947	0.961	0.953	0.938	32	0.699	0.872	0.788	0.585
a-RotatE	309	0.949	0.959	0.952	0.944	40	<u>0.797</u>	0.884	0.830	<u>0.746</u>
QuatE	<u>162</u>	0.950	0.959	<u>0.954</u>	<u>0.945</u>	17	0.782	0.900	<u>0.835</u>	0.711
DualE ¹	-	<u>0.951</u>	0.961	0.956	<u>0.945</u>	-	0.790	0.881	0.829	0.734
DualE ²	156	0.952	<u>0.962</u>	0.956	0.946	<u>21</u>	0.813	<u>0.896</u>	0.850	0.766

Table 2: Link prediction results on WN18 and FB15K. Best results are in bold and second best results are underlined.

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

Table 3: Number of entities, relations, and observed triplets in each split for four benchmarks.

is to model and infer the **symmetry/antisymmetry** and **inversion** patterns. WN18RR (Dettmers et al. 2017) is a subset of WN18. The inverse relations are deleted, and the main relation patterns are **symmetric/antisymmetric** and **composition**. FB15K-237 (Toutanova and Chen 2015) is a subset of FB15K, where the inverse relations are deleted. Therefore, the key to link prediction on FB15K-237 boils down to model and infer **symmetrical/antisymmetric** and **composition** patterns.

Baselines: We compare our method DualE to state-of-the-art models, including MurP (Balažević, Allen, and Hospedales 2019), MurE (which is the Euclidean analogue of MurP), RotatE (Sun et al. 2019), ComplEx-N3 (Lacroix, Usunier, and Obozinski 2018), ROTH (Chami et al. 2020) and TuckER (Balažević, Allen, and Hospedales 2019). Some neural network-based methods even use dropout and label smoothing to improve their performance. For DualE, the number of negative samples are 10(WN18), 10 (FB15K), 2(WN18RR), 10(FB15K-237), respectively.

Evaluation Protocol: In this paper, we use Mean Ranking (MR), Mean Reciprocal Ranking (MRR) and Hits@n as evaluation indicators. MR can indicate performance measured by the average rank of all correct entities with lower values. And the reverse ranking of the correct entity is represented by MRR. Then we measure the proportion of the first n correct entities by Hits@n, where $n = 1, 3, 10$. We refer to (Bordes

et al. 2013) for the processing of the original results, and the filtered results are shown in results.

Implementation Details: Our settings for hyper-parameter selection using pytorch are as follows: The embedding size k is selected in $\{50, 100, 150, 200, 250\}$. The regularization rates λ_1 and λ_2 are adjusted in $\{0, 0.02, 0.05, 0.1, 0.15, 0.2\}$. The learning rate is chosen from 0.02 to 0.1, and different learning rates can be selected according to different datasets. In addition, we create 10 batches of training samples for all the datasets above. For details about the epoch we use, please refer to Table 6. Note that in order to ensure a fair comparison, we use fewer dimensions than other baselines, so that the number of parameters remains the same.

Results: From Table 2 and Table 4, we can see that DualE surpasses other state-of-art models to achieve the best performance. We use two versions of DualE in the experiment, which is without/with type constraints (Krompa, Baier, and Tresp 2015) called DualE¹ and DualE². Next, we conduct a detailed analysis of each dataset.

On WN18, we achieve the best performance, which shows that DualE can learn symmetry/antisymmetry and inversion patterns well. The main relations included in the FB15K dataset are similar to that of WN18. The performance of DualE is equal to that of QuatE in MR and hits@10, but it obtains obvious advantages in MRR, hits@3 and hits@1. On WN18RR, TransE cannot learn the symmetric relation pattern, so it performs not well. In contrast, the rotation family can achieve good results, and DualE has further refreshed the performance to achieve the optimal. On FB15K-237, the performance of DualE can be improved by several percentage points compared with the previous state-of-the-art models, which shows that DualE can learn the composition relation pattern better. We have also noticed some competitive models such as MURP and ROTH, which extend the embedding space to hyperbolic space while they model the relation as a single translation or rotation. But their experimental results on FB15K237 are worse than DualE, which fully demonstrate

Model	WN18RR					FB15K-237				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	3384	0.226	0.501	-	-	357	0.294	0.465	-	-
DistMult \diamond	5100	0.43	0.49	0.44	0.39	254	0.241	0.419	0.263	0.155
ComplEx \diamond	5261	0.44	0.51	0.46	0.41	339	0.247	0.428	0.275	0.158
ConvE \diamond	4187	0.43	0.52	0.44	0.40	244	0.325	0.501	0.356	0.237
R-GCN+	-	-	-	-	-	-	0.249	0.417	0.264	0.151
NKGE	4170	0.45	0.526	0.465	0.421	237	0.33	0.510	0.365	0.241
RotatE \spadesuit	3277	0.470	0.565	0.488	0.422	185	0.297	0.480	0.328	0.205
a-RotatE \spadesuit	3340	0.476	0.571	0.492	0.428	177	0.338	0.533	0.375	0.241
QuatE	<u>2363</u>	<u>0.491</u>	0.579	<u>0.510</u>	0.441	88	0.352	<u>0.555</u>	0.392	0.258
ComplEx-N3	-	0.480	0.572	<u>0.495</u>	0.435	-	0.357	<u>0.547</u>	0.392	0.264
TuckER	-	0.470	0.526	0.482	<u>0.443</u>	-	<u>0.358</u>	0.544	<u>0.394</u>	<u>0.266</u>
MURP	-	0.475	0.554	0.487	0.436	-	0.336	0.521	0.370	0.245
ROTH	2293	<u>0.491</u>	0.586	0.511	0.441	-	0.344	0.535	0.380	0.246
DualE¹	-	0.482	0.561	0.500	0.440	-	0.330	0.518	0.363	0.237
DualE²	2270	0.492	<u>0.584</u>	0.513	0.444	<u>91</u>	0.365	0.559	0.400	0.268

Table 4: Link prediction results on WN18RR and FB15K-237. [\diamond]: Results are taken from (Dettmers et al. 2017); [\spadesuit]: Results are taken from (Sun et al. 2019).

the superiority of the combination of translation and rotation.

We also notice that QuatE apply N3 regularization and reciprocal learning approaches (Lacroix, Usunier, and Obozinski 2018) to improve performance. It mentions that using N3 and reciprocal learning on FB15K and FB15K-237 could boost the performances well. But it requires a large embedding dimension We think it is not preferable since our original intention is embedding entities and relations to a lower dimensional space. In contrast, the effect of DualE model in fewer dimensions can reach and surpass that of QuatE model with the N3 regularization.

Experimental Analysis

Ablation Study on Dual Quaternion Normalization. We remove the normalization step in DualE and use the original relation quaternion W_r to project head entity, but it makes the results poorer. It is likely because scaling effects in non-unit dual quaternions are detrimental. We also add more normalization steps in DualE to study the impact of normalization. Recall normalization steps we use before, we add the following normalization base on Eq.(3): $d' = \frac{d}{\|d\|} = \frac{d_0 + d_1 i + d_2 j + d_3 k}{\sqrt{d_0^2 + d_1^2 + d_2^2 + d_3^2}}$. Then we denote

$W_r^\dagger = (c', d') = (c, d) = (c_0, c_1, c_2, c_3, d_0, d_1, d_2, d_3)$. We also do dual quaternion multiplication between head and tail dual quaternions and consider the relation quaternion as weight. Thus we have $\phi(h, r, t) = \langle W_r, Q_h \otimes Q_t \rangle$, which will lead the poor performance, for the reason is that the geometric property of relational translation and rotation is lost. The results are shown in Table 7.

Impact of embedding dimension. To understand the role of dimensionality, we also conduct experiments on WN18RR against SotA methods under varied low-dimensional settings. As shown in Figure 4, our approach consistently outperforms all baselines, suggesting that DualE still attains high-accuracy

across a broad range of dimensions.

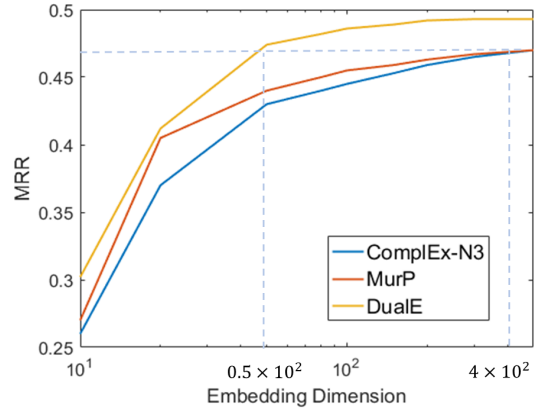


Figure 4: Impact of embedding dimension.

Conclusion

In order to overcome the shortcomings of previous knowledge graph embeddings, we design a new knowledge graph embedding model called DualE. Based on dual quaternions, DualE can model the relation as both rotation and translation. The dual quaternion space has clear mathematical and physical meanings and all key relation patterns and multiple relations pattern can be modeled. We have also proved that DualE model is the unified framework of rotation family and translation family, combining the two branches of the research on modeling relations as rotation or translation. The experimental evaluations on datasets show that our DualE outperforms other state-of-the-art methods.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under (Grant No. 2018AAA0102003), in part by National Natural Science Foundation of China (61620106009, 61861166002, U1636214, 61733007, 61931008, 61836002, 61672514, and 61976202), in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Beijing Education Committee Cooperation Beijing Natural Science Foundation (No.KZ201910005007), in part by Beijing Natural Science Foundation (No. 4182079), in part by Youth Innovation Promotion Association CAS, and in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000.

References

- Balažević, I.; Allen, C.; and Hospedales, T. M. 2019. Tucker: Tensor factorization for knowledge graph completion. In *EMNLP*, 5185–5194.
- Bansal, T.; Juan, D.-C.; Ravi, S.; and McCallum, A. 2019. A2N: attending to neighbors for knowledge graph inference. In *ACL*, 4387–4392.
- Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD*, 1247–1250.
- Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2014. A semantic matching energy function for learning with multi-relational data Application to word-sense disambiguation. *Machine Learning* 94(2): 233–259.
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*, 2787–2795.
- Bordes, A.; Weston, J.; Collobert, R.; and Bengio, Y. 2011. Learning structured embeddings of knowledge bases. In *national conference on artificial intelligence*, 301–306.
- Chami, I.; Wolf, A.; Juan, D.-C.; Sala, F.; Ravi, S.; and Ré, C. 2020. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In *ACL*.
- Clifford. 1871. Preliminary Sketch of Biquaternions. *Proceedings of The London Mathematical Society* (1): 381–395.
- Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2017. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*, 1811–1818.
- Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research* 12(Jul): 2121–2159.
- Ebisu, T.; and Ichise, R. 2018. TorusE: Knowledge Graph Embedding on a Lie Group. In *AAAI*, 1819–1826.
- Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 249–256.
- Hamilton, W. R. 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.
- He, S.; Liu, K.; Ji, G.; and Zhao, J. 2015. Learning to represent knowledge graphs with gaussian embedding. In *CIKM*, 623–632.
- Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *ACL*, 687–696.
- Ji, G.; Liu, K.; He, S.; and Zhao, J. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *AAAI*, 985–991.
- Krompa, D.; Baier, S.; and Tresp, V. 2015. Type-Constrained Representation Learning in Knowledge Graphs. In *ISWC*.
- Lacroix, T.; Usunier, N.; and Obozinski, G. 2018. Canonical Tensor Decomposition for Knowledge Base Completion. In *ICML*, 2863–2872.
- Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 2181–2187.
- Miller, G. 1995. Wordnet: a lexical database for english communications of the acm 38 (11) 3941. *Niemela, I*.
- Nathani, D.; Chauhan, J.; Sharma, C.; and Kaul, M. 2019. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. In *ACL*, 4710–4723.
- Nguyen, D. Q.; Nguyen, T. D.; Nguyen, D. Q.; and Phung, D. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *NAACL-HLT*, 327–333.
- Nguyen, D. Q.; Sirts, K.; Qu, L.; and Johnson, M. 2016. STTransE: a novel embedding model of entities and relationships in knowledge bases. In *ACL*, 460–466.
- Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic Embeddings of Knowledge Graphs. In *AAAI*, 1955–1961.
- Parcollet, T.; Ravanelli, M.; Morchid, M.; Linarès, G.; Traubelsi, C.; De Mori, R.; and Bengio, Y. 2018. Quaternion recurrent neural networks. In *ICLR*.
- Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *ESWC*, 593–607. Springer.
- Sun, Z.; Deng, Z.; Nie, J.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*, 1–18.
- Sutskever, I.; Tenenbaum, J. B.; and Salakhutdinov, R. 2009. Modelling Relational Data using Bayesian Clustered Tensor Factorization. In *NIPS*, 1821–1828.
- Toutanova, K.; and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *CVSC*, 57–66.
- Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *ICML*, 2071–2080.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 1112–1119.

Xiao, H.; Huang, M.; Hao, Y.; and Zhu, X. 2015. TransA: An Adaptive Approach for Knowledge Graph Embedding. In *AAAI*, 1–7.

Xiao, H.; Huang, M.; and Zhu, X. 2016. TransG: A generative model for knowledge graph embedding. In *ACL*, 2316–2325.

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*, 1–13.

Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q. 2019. Quaternion knowledge graph embeddings. In *NIPS*, 2731–2741.