Clifford Embeddings – A Generalized Approach for Embedding in Normed Algebras

Caglar Demir[0000−0001−8970−3850] (✉) and Axel-Cyrille Ngonga Ngomo[0000−0001−7112−3516]

Data Science Research Group, Paderborn University, Germany
{caglar.demir@, axel.ngonga@}upb.de

Abstract. A growing number of knowledge graph embedding models exploit the characteristics of division algebras (e.g., $\mathbb{R}$, $\mathbb{C}$, $\mathbb{H}$, and $\mathbb{O}$) to learn embeddings. Yet, recent empirical results suggest that the suitability of algebras is contingent upon the knowledge graph being embedded. In this work, we tackle the challenge of selecting the algebra within which a given knowledge graph should be embedded by exploiting the fact that Clifford algebras $Cl_{p,q}$ generalize over $\mathbb{R}$, $\mathbb{C}$, $\mathbb{H}$, and $\mathbb{O}$. Our embedding approach, Keci, is the first knowledge graph embedding model that can parameterize the algebra within which it operates. With Keci, the selection of an underlying algebra becomes a part of the learning process. Specifically, Keci starts the training process by learning real-valued embeddings for entities and relations in $\mathbb{R}^m = Cl^m_{0,0}$. At each mini-batch update, Keci can steer the training process from $Cl^m_{p,q}$ to $Cl^m_{p+1,q}$ or $Cl^m_{p,q+1}$ by processing the training loss. In this way, Keci can decide the algebra within which it operates in a data-driven fashion. Consequently, Keci is a generalization of previous approaches such as DistMult, ComplEx, QuatE, and OMult. Our evaluation suggests that Keci outperforms state-of-the-art embedding approaches on seven benchmark datasets. We provide an open-source implementation of Keci, including pre-trained models, training and evaluation scripts.[1]

Keywords: Knowledge Graphs · Embeddings · Theory Unification

1 Introduction

A plethora of knowledge graph embedding (KGE) models have been developed over the last decade [34, 7, 33]. Most KGE models map entities $e \in \mathcal{E}$ and relations $r \in \mathcal{R}$ found in a knowledge graph (KG) $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ to $\mathbb{V}$, where $\mathbb{V}$ is a d-dimensional vector space and $d \in \mathbb{N}\backslash\{0\}$ [17]. This family of models is currently one of the most popular means to make KGs amenable to vectorial machine learning [33] and has been used in applications including drug discovery, community detection, recommendation, question answering [15, 34, 14, 3]. While early models (e.g., RESCAL [24], DistMult [37]) express embeddings in $\mathbb{R}^d$ and perform well once tuned fittingly [28], later results suggest that embedding using

[1] https://github.com/dice-group/dice-embeddings

the more complex division algebras $\mathbb{C}$ and $\mathbb{H}$ can achieve a superior link prediction performance (measured in terms of hits at n, short h@n) [28, 42, 8, 40]. This is at least partially due to the characteristics of (hyper)complex algebras (e.g., $\mathbb{C}$, $\mathbb{H}$) being used to account for logical properties such as the symmetry, asymmetry, and compositionality [30] of relations r found in the input data. While recent works have continued improving the performance of KGE models by including ever more complex neural architectures atop division algebras (see, e.g., ConvE [12], ConEx [9]), a fundamental assumption remains shared: $\mathbb{V}$ is fixed for each approach. This assumption has the advantages of being conducive to rapid implementation, execution and interpretation. However, its main disadvantage is well illustrated by experimental results from recent works [28]: The most adequate algebra for embedding a KG is contingent upon the data to embed. This finding is corroborated by our experimental results (see Figure 1).

The link between algebras and KGEs can be directly entailed from formal treatments on embeddings. Consider ComplEx [32] embeddings for example. When a KG does not contain triples with an anti-symmetric relation (e.g., the `bornIn` or `hasChild` relations), embedding into $\mathbb{C}$ instead of $\mathbb{R}$ is of no advantage. On the contrary, the space and time requirements of an approach based on $\mathbb{C}$ are double that of an approach based on $\mathbb{R}$ (see Section 4). Similar insights can be derived for the division algebras $\mathbb{H}$ and $\mathbb{O}$ (see, e.g., QuatE [42] and ConvO [8]). Our goal therefore is to find a hypothesis space, i.e., a suitable algebra underlying $\mathbb{V}$ that is rich enough to embed the input knowledge graph, yet simple enough to ensure reliable generalization over unseen data. We implement this goal by presenting Keci. Our approach exploits the fact that Clifford Algebras $Cl_{p,q}$ (see Section 2.1 for an overview) generalize over common division algebras and aims to find a suitable algebra for a particular dataset in a data-driven fashion. We conceived of two ways to find a suitable algebra: (1) Consider p and q as two new hyperparameters to find a suitable $Cl_{p,q}$ or (2) scale the imaginary dimensions of $Cl_{p,q}$ according to the training loss. While finding a suitable $Cl_{p,q}$ via (1) is computationally expensive as it requires multiple training phases, (2) is conceptually simpler. Keci is hence based on (2), and uses the cross-entropy training loss to decide whether it should increase the number of imaginary dimensions of the division algebra within which it operates. This process can be equated with starting the learning process in a small hypothesis space (e.g., $\mathbb{R} = Cl_{0,0}$) and steering the learning process towards a larger hypothesis space as required. Our experiments on seven benchmark datasets (WN18RR, FB15K-237, YAGO3-10, NELL-995-h25, NELL-995-h50, NELL-995-h75, UMLS, and KINSHIP) suggest that Keci outperforms DistMult, ComplEx, QMult and OMult across all datasets and benchmark metrics (MRR, Hit@1,Hit@3, and Hit@10).

2 Preliminaries and Notation

2.1 Clifford Algebras

A Clifford algebra $Cl_{p,q}(\mathbb{R})$ is an associative algebra (i.e., additions and multiplications are associative) generated by the $p + q$ orthonormal basis elements

$e_1, \ldots, e_{p+q}$ for which the following relations hold:

$$e_i^2 = +1 \quad \text{for} \ \ 1 \leq i \leq p \ , \tag{1}$$

$$e_j^2 = -1 \quad \text{for} \ \ p < j \leq p+q \ , \tag{2}$$

$$e_i e_j = -e_j e_i \quad \text{for} \ \ i \neq j \ . \tag{3}$$

The lowest-dimensional Clifford algebra $Cl_{0,0}(\mathbb{R})$ is a zero-dimensional algebra with vector space $\mathbb{V}$ that is spanned by the basis element $\{1\}$. Hence, $Cl_{0,0}(\mathbb{R})$ is algebra-isomorphic to $\mathbb{R}$. Analogously, $Cl_{0,1}(\mathbb{R})$ is equivalent to $\mathbb{C}$, and $Cl_{0,2}(\mathbb{R})$ is equivalent to $\mathbb{H}$ [16, 5]. In fact, the spaces used by a large portion of the state-of-the-art KGE models are (sub-) algebras of some $Cl_{p,q}(\mathbb{R})$ (see Section 3 and Table 1).

2.2 Knowledge Graphs

A KG represents structured collections of assertions describing the world [17]. Formally, a KG is often defined as a set of triples $\mathcal{G} := \{(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$, where $\mathcal{E}$ and $\mathcal{R}$ stand for a set of entities and a set of relations, respectively [12, 2, 1]. Each triple $(h, r, t) \in \mathcal{G}$ represents an assertion based on two entities $h, t \in \mathcal{E}$ and a relation $r \in \mathcal{R}$. A relation r is *symmetric* iff $(h, r, t) \iff (t, r, h)$ holds. Analogously, r is *anti-symmetric* iff $(h, r, t) \in \mathcal{G} \Rightarrow (t, r, h) \notin \mathcal{G}$ for all $h \neq t$. Most publicly available KGs contain missing and erroneous assertions [17]. These triples can be inferred from an existing set of triples by means of designing logical rules or learning continuous vector representations via knowledge graph embedding models [22].

2.3 Knowledge Graph Embeddings

Most KGE models learn continuous vector representations tailored towards link prediction [6, 17]. They are often defined as parameterized scoring functions $\phi_\Theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \mapsto \mathbb{R}$, where Θ denotes parameters and often comprise entity embeddings $\mathbf{E} \in \mathbb{V}^{|\mathcal{E}| \times d_e}$, relation embeddings $\mathbf{R} \in \mathbb{V}^{|\mathcal{R}| \times d_r}$, and additional parameters (e.g., affine transformations, batch normalizations, convolutions) [1, 8]. Since $d_e = d_r$ holds for many models including models reported in Table 1, we will use d to signify the number of real parameters used for the embedding of an entity or relation Given $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, the prediction $\hat{y} := \phi_\Theta(h, r, t)$ signals the likelihood of (h, r, t) being true [12, 33, 41]. Since $\mathcal{G}$ contains only assertions that are assumed to be true, assertions assumed to be false are often generated by applying the negative sampling, 1vsAll or Kvsall training strategies [28]. Throughout this paper, we will denote embeddings with bold fonts, i.e., the embedding of h will be denoted $\mathbf{h}$. Moreover, we use $\circ$ and $\cdot$ to denote an element-wise vector multiplication and an inner product in $\mathbb{V}$, respectively.

3 Related Work

In the last decade, a plethora of KGE models have been successfully applied to tackle various tasks, including link prediction, class expression learning, drug discovery among many others [12, 33, 28, 11, 38, 25, 39, 3]. Most KGE models are designed to operate in a pre-determined vector space $\mathbb{V}$ based on a normed division algebra to learn embeddings for entities and relations tailored towards predicting missing links. Most of these models can be unified under *a feature composition operator* followed by *an approximation operator* in a respective division algebra. Given a triple (h, r, t), most KGE models $\phi_\Theta : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \mapsto \mathbb{R}$ computes a triple score via linear operations (e.g., element-wise multiplications or additions) on $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ [4, 40, 37, 42, 8, 32].

3.1 Inner Product vs Distance

A large portion of the existing KGE approaches can be regarded as instances of one of two paradigms. RESCAL [24], DistMult [37], ComplEx [32], ComplEx-N3 [19], QuatE, OctonionE [42], QMult, and OMult [8] can be unified under the inner product paradigm, where an inner product is used as an approximation operator in a preselected vector space $\mathbb{V}$. TransE [4], TransH [35], TransR [20], CTransR [20], TransD [18], TransO [40], and RotatE [29] can be regarded as belonging to the distance paradigm, where a distance (e.g., the Euclidean distance) is used as an approximation operator in a selected vector space $\mathbb{V}$. Given a triple (h, r, t), all aforementioned models apply element-wise multiplication or addition to obtain a composite representation of the head entity embedding $\mathbf{h}$ and the relation embedding $\mathbf{r}$ in $\mathbb{V}$. A scalar real-valued prediction $\hat{y} := \phi_\Theta(h, r, t)$ is obtained via an inner product or a distance between a resulting composite representation and the tail entity embedding $\mathbf{t}$.

Table 1. State-of-the-art embedding approaches and algebras used for embeddings

Models	**Vector Space**	$\subseteq Cl_{p,q}(\mathbb{R})$
TransE, DistMult, RESCAL	$\mathbb{R}$	$Cl_{0,0}$
ComplEx, RotatE, ConEx	$\mathbb{C}$	$Cl_{0,1}$
QuatE, QMult, DensE	$\mathbb{H}$	$Cl_{0,2}$
OMult, OctonionE	$\mathbb{O}$	$Cl_{1,3}$

To obtain a more expressive composite representation of $\mathbf{h}$ and $\mathbf{r}$, various KGE models (e.g., HolE [23], ConvE [12], HypER [1], ConvKB [21], ConEx [9], ConvQ [8], ConvO [8] and AcrE [27]) apply 1D or 2D convolutions followed by an non-linear affine transformation as a feature composition operator. These convolution-based models aim to learn a complex composite representation of h and r that is ideally approximately equal to $\mathbf{t}$, while maintaining a parameter efficiency.

3.2 Selecting $\mathbb{V}$

The vector space $\mathbb{V}$ can encode useful prior knowledge that enable KGE models to infer missing triples. For instance, given a triple (h, r, t), DistMult computes a triple score via $(\mathbf{h} \circ \mathbf{r}) \cdot \mathbf{t}$, where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$, $\circ$ denotes the element-wise vector multiplication, and $\cdot$ stands for the inner product in $\mathbb{V} = \mathbb{R}$. This formulation leads DistMult to enjoy the linear time and space complexity of multiplication and inner product in $\mathbb{R}$. However, although DistMult can accurately infer missing triples with symmetric relations, missing triples with anti-symmetric relations cannot be accurately inferred. To alleviate this shortcoming and retain parameter and computational efficiency, ComplEx extends DistMult into $\mathbb{C}$, where $\circ$ denotes the element-wise complex vector multiplication and $\cdot$ is a Hermitian inner product. Since this inner product in $\mathbb{C}$ is symmetric in $\mathrm{Re}(\cdot)$ and anti-symmetric in $\mathrm{Im}(\cdot)$, triples with anti-symmetric relations can be accurately predicted [31].

Overall, recent empirical results suggest that the suitability of an algebra is contingent upon the input KG that is to be embedded. For instance, ComplEx outperforms DistMult by absolute 0.5 % and 2.2% MRR scores on FB15K-237 and WN18RR, respectively, provided that the models are well tuned and $d = 256$ [28]. On the other hand, a recent work shows that DistMult outperforms ComplEx by 2.1% absolute MRR on FB15K-237, while ComplEx outperforms DistMult by 2.4% and 0.5% absolute MRR on WN18RR and YAGO3-10, respectively. Note that in this experiment, DistMult operates on $\mathbb{R}^{100}$ and ComplEx on $\mathbb{C}^{50}$ [9].[2] Similarly, another recent work suggests that in a low-dimensional setting with $d = 32$, DistMult outperforms ComplEx by 3.5% and 3.2% absolute MRR on FB15K-238 and NELL-995-h100, respectively, while ComplEx outperforms DistMult by 3.6% absolute MRR score on WN18RR [13]. Another recent clinical study pertaining to drug discovery with $d = 200$ shows that the recall@200 performance of ComplEx is 5.7% higher than the performance of DistMult [25]. Many other recent works–including [28, 25, 9, 13]–indicate similar dependencies of the suitability of an algebra on the input KG. On the application side, studies such as Bonner et al. [3] show that determining the selection criteria of a particular KGE model over another real datasets is a viable question. They highlight the importance of understanding the properties of such models w.r.t. the input dataset to improve drug discovery efforts.

We argue that the selection of the algebra underlying the vector space $\mathbb{V}$ within which a KGE model learns embeddings for a given KGE can be a part of the learning problem. In Section 4, we introduce the first KGE model that does not only learn embeddings for entities and relations but also a suitable algebra for $\mathbb{V}$ by means of a dimension scaling technique.

4 Methodology

4.1 Clifford Embeddings

Given a triple $(h, r, t) \in \mathcal{G}$, let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in Cl_{p,q}(\mathbb{R}^m)$ denote three multi-vectors representing embeddings of the head entity h, the relation r and the tail entity

[2] Note that the two models have the same complexity w.r.t. the number of real numbers necessary to represent the final embeddings as every element of $\mathbb{C}$ is encoded via two real numbers.

t, respectively. We defined the multivector $\mathbf{h}$ as

$$\mathbf{h} = h_0 + \sum_{i=1}^{p} h_i e_i + \sum_{j=p+1}^{p+q} h_j e_j, \tag{4}$$

where $h_{(\cdot)} \in \mathbb{R}^m$ with $m = \lfloor d/(p+q+1) \rfloor$. The vectors $\mathbf{r}$ and $\mathbf{t}$ are defined analogously. Hence, every embedding vector can be represented by at most d real numbers. The Clifford multiplication of $\mathbf{h}$ and $\mathbf{r}$ is given by

$$\begin{aligned}
\mathbf{h} \circ \mathbf{r} = & h_0 r_0 && + \sum_{i=1}^{p} h_0 r_i e_i && + \sum_{j=p+1}^{p+q} h_0 r_j e_j && (5)\\
& + \sum_{i=1}^{p} h_i r_0 e_i && + \sum_{i=1}^{p} \sum_{k=1}^{p} h_i r_k e_i e_k && + \sum_{i=1}^{p} \sum_{j=p+1}^{p+q} h_i r_j e_i e_j && (6)\\
& + \sum_{j=p+1}^{p+q} h_j r_0 e_j && + \sum_{j=p+1}^{p+q} \sum_{i=1}^{p} h_j r_i e_j e_i && + \sum_{j=p+1}^{p+q} \sum_{k=p+1}^{p+q} h_j r_k e_j e_k. && (7)
\end{aligned}$$

Grouping the terms using the bases and applying the Clifford algebra bases rules (see Section 2.1) simplifies the above expression to

$$\mathbf{h} \circ \mathbf{r} = \sigma_0 + \sigma_p + \sigma_q + \sigma_{p,p} + \sigma_{q,q} + \sigma_{p,q}, \tag{8}$$

where $\sigma_{(\cdot)}$ are defined as

$$\sigma_0 = h_0 r_0 + \sum_{i=1}^{p} h_i r_i - \sum_{j=p+1}^{p+q} h_j r_j, \tag{9}$$

$$\sigma_p = \sum_{i=1}^{p} (h_0 r_i + h_i r_0) e_i, \tag{10}$$

$$\sigma_q = \sum_{j=p+1}^{p+q} (h_0 r_j + h_j r_0) e_j, \tag{11}$$

$$\sigma_{p,p} = \sum_{i=1}^{p-1} \sum_{k=i+1}^{p} (h_i r_k - h_k r_i) e_i e_k, \tag{12}$$

$$\sigma_{q,q} = \sum_{j=1}^{p+q-1} \sum_{k=j+1}^{p+q} (h_j r_k - h_k r_j) e_j e_k, \tag{13}$$

$$\sigma_{p,q} = \sum_{i=1}^{p} \sum_{j=p+1}^{p+q} (h_i r_j - h_j r_i) e_i e_j. \tag{14}$$

4.2 Scoring Function based on Inner Product

Given a triple (h, r, t) and the respective multi-vector embeddings of $\mathbf{h}, \mathbf{r}, \mathbf{t} \in Cl_{p,q}(\mathbb{R}^m)$, Keci's scoring function is given by

$$\text{Keci}(h, r, t)_{p,q} = (\mathbf{h} \circ \mathbf{r}) \cdot \mathbf{t}, \tag{15}$$

where $\cdot$ denotes the inner product between two multi-vectors in $Cl_{p,q}(\mathbb{R}^m)$. Distributing the components of $\mathbf{t}$ simplifies to

$$\begin{aligned}
\text{Keci}(h, r, t)_{p,q} =& h_0 r_0 t_0 + \sum_{i=1}^{p} (h_i r_i t_0) - \sum_{j=p+1}^{p+q} (h_j r_j t_0) && (16)\\
&+ \sum_{i=1}^{p} (h_0 r_i t_i + h_i r_0 t_i) e_i && (17)\\
&+ \sum_{j=p+1}^{p+q} (h_0 r_j t_j + r_0 h_j t_j) e_j && (18)\\
&+ \sigma_{p,p} + \sigma_{q,q} + \sigma_{p,q}. && (19)
\end{aligned}$$

Therefore, Keci can be classified as a knowledge graph embedding model using element-wise multiplication as a feature composition operator and the inner product as an approximation operator. Hence, Keci is akin to DistMult, ComplEx, QMult and OMult depending on p and q. More specifically, selecting $p = q = 0$ leads $\text{Keci}(h, r, t)_{0,0}$ to generalize to DistMult:

$$\text{Keci}(h, r, t)_{p,q} = h_0 \circ r_0 \cdot t_0 = \langle \text{Re}(\mathbf{h}), \text{Re}(\mathbf{r}), \text{Re}(\mathbf{t}) \rangle, \tag{20}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in Cl_{0,0}(\mathbb{R}^d)$, hence, $h_0, r_0, t_0 \in \mathbb{R}^{m=d}$. Similarly, selecting $p = 0 \wedge q = 1$ leads to $\text{Keci}(h, r, t)_{0,1}$, which is equivalent to ComplEx.

$$\begin{aligned}
\text{Keci}(h, r, t)_{p,q} = \mathbf{h} \circ \mathbf{r} \cdot \mathbf{t} =& \langle \text{Re}(\mathbf{h}), \text{Re}(\mathbf{r}), \text{Re}(\mathbf{t}) \rangle && (21)\\
&+ \langle \text{Re}(\mathbf{h}), \text{Im}(\mathbf{r}), \text{Im}(\mathbf{t}) \rangle && (22)\\
&+ \langle \text{Im}(\mathbf{h}), \text{Re}(\mathbf{r}), \text{Im}(\mathbf{t}) \rangle && (23)\\
&- \langle \text{Im}(\mathbf{h}), \text{Im}(\mathbf{r}), \text{Re}(\mathbf{t}) \rangle, && (24)
\end{aligned}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^{\frac{d}{2}}$. This clearly shows that the triple score computed by Keci does not only depend on learned embeddings of h, r, t but also parameterization of an algebra.

4.3 Learning to scale dimensions in $Cl_{p,q}(\mathbb{R})$

Remember that we argue that the selection of the vector space within which a KGE operates should be a part of the learning problem. Hence, instead of fixing p and q for Keci, we argue that Keci should be endowed with the capability of selecting a particular subspace of $Cl_{p,q}$ to operate in.

We propose to learn coefficients $\alpha_1, \dots \alpha_{p+q}$ for for the orthonormal bases $e_1, \dots e_{p+q}$ of $Cl_{p,q}(\mathbb{R})$, where $\alpha_i \in \mathbb{R}$. Entity embeddings and relation embeddings are now of the forms

$$\mathbf{h}_\alpha = h_0 + \sum_{i=1}^{p} h_i \alpha_i e_i + \sum_{j=p+1}^{p+q} h_j \alpha_j e_j, \text{ and} \tag{25}$$

$$\mathbf{r}_\alpha = r_0 + \sum_{i=1}^{p} r_i \alpha_i e_i + \sum_{j=p+1}^{p+q} r_j \alpha_j e_j, \tag{26}$$

where $\alpha_i, \alpha_j \in \mathbb{R}$ are trainable coefficients for each base vectors. Initializing $\alpha_1, \dots, \alpha_{p+q} = 0$ leads Keci to start the training process as DistMult in $\mathbb{R}^m$, where $m = \lfloor d/(p+q+1) \rfloor$. Hence, for a given (h, r, t), a triple score is computed as

$$\mathbf{h}_\alpha \circ \mathbf{h}_\alpha \cdot \mathbf{t}_\alpha = h_0 \circ r_0 \cdot t_0, \tag{27}$$

where the valued represented along e_1, e_{p+q} are scaled down to 0. During training, $\alpha_1, \dots, \alpha_{p+q}$ can be updated iteratively on the basis of the training loss. More specifically, let $\mathcal{L}$ denote the cross-entropy loss function that is defined as

$$\mathcal{L}(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y}), \tag{28}$$

where y denotes a binary label of a given triple (h, r, t) and $\hat{y} = \sigma(\phi_\Theta(h, r, t))$ denotes a prediction obtained via the logistic sigmoid function ($\sigma(x) = \frac{1}{1+e^{-x}}$). Moreover, let $\frac{d\mathcal{L}}{d\alpha_i}$ denote the derivative of $\mathcal{L}$ w.r.t. α_i on a single data point (h, r, t). Therefore, a coefficient α_i is updated on the bases of h_i and r_i, $\frac{d\mathcal{L}}{dh_i \alpha_i} h_i + \frac{d\mathcal{L}}{dr_i \alpha_i} r_i$. In the mini-batch training setting, α_i is updated with a batch of respective terms. Consequently, updating α_i in the negative direction of the gradients assists to decrease the loss further. Hence, Keci can learn to select a particular subspace of $Cl_{p,q}$ that is more favorable to decrease the training loss.

5 Experiments

5.1 Datasets

We used the benchmark datasets UMLS, KINSHIP, NELL-995 h25, NELL-995 h50, NELL-995 h100, FB15K-237, and YAGO3-10 for the link prediction problem. An overview of the datasets is provided in Table 2. UMLS describes relationships between medical entities and their relationships, e.g., `immunologic_factor`, `disrupts`, and `cell`. KINSHIP describes the 25 different kinship relations of the Alyawarra tribe and UMLS describes 135 medical entities via 46 relations describing [30]. FB15K-237 and YAGO3-10 are subsets of Freebase and YAGO [12]. They contain information about a general domain, e.g., `Stephen_Hawking`, and `Copley_Medal`. The Never-Ending Language Learning datasets NELL-995 h25, NELL-995 h50, and NELL-995 h100 are designed to evaluate multi-hop reasoning capabilities [36].

Table 2. An overview of datasets in terms of number of entities, number of relations, and node degrees in the train split along with the number of triples in each split of the dataset.

Dataset	$\|\mathcal{E}\|$	$\|\mathcal{R}\|$	$\|\mathcal{G}^{\text{Train}}\|$	$\|\mathcal{G}^{\text{Validation}}\|$	$\|\mathcal{G}^{\text{Test}}\|$
UMLS	135	46	5,216	652	661
KINSHIP	104	25	8,544	1,068	1,074
NELL-995 h100	22,411	43	50,314	3,763	3,746
NELL-995 h50	34,667	86	72,767	5,440	5,393
NELL-995 h25	70,145	172	122,618	9,194	9,187
FB15K-237	14,541	237	272,115	17,535	20,466
YAGO3-10	123,182	37	1,079,040	5,000	5,000

5.2 Experimental Setup and Optimization

Throughout our experiments, we use the cross-entropy loss function to train each knowledge graph embedding model. We evaluated the link prediction performance of models with benchmark metrics (filtered MRR, Hits@1,Hits@3, and Hits@10). We did not used any regularization technique (e.g., dropout technique or L2 regularization) as we report the training and validation performance of each model on each dataset. Throughout our experiments, each entity and relation is represented with 32-dimensional real valued vector across datasets and models as in [6, 13]. Hence, DistMult, ComplEx, QMult, and OMult learn embeddings in $\mathbb{R}^{32}, \mathbb{C}^{16}, \mathbb{H}^{8}$, and $\mathbb{O}^{4}$, respectively. Consequently, all models have the same number of parameters. We report the training, validation and test results to prove a finer-grained overview of performance across datasets and models. We use the Adam optimizer with 0.1 learning rate and train each model for 256 epochs with the batch size of 1024. The implementation of Keci can be found in the dice-embedding framework [10]. Therein, we also provided the pre-trained models[3].

6 Results

6.1 Exhaustive Search

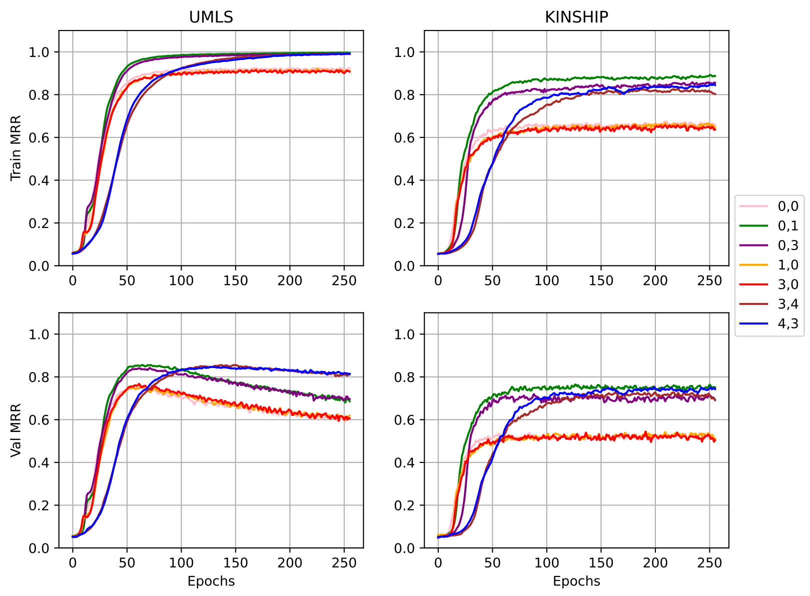
The goal of our first series of experiments was to verify that the performance of Keci is contingent upon different values for p and q. Hence, we did not perform any dimension scaling and tried all combinations of $p \leq 4$ and $q \leq 4$. Note that we only kept one copy of (p, q) pairs for each equivalent class of $Cl_{p,q}$. For example, $Cl_{1,1}$ is isomorphic to $Cl_{0,2}$.

Figure 1 shows the MRR trajectories of Keci with different (p, q) pairs on the UMLS and KINSHIP benchmark datasets. At the end of each training epoch, the MRR performances on the training and validation splits was registered and

[3] https://github.com/dice-group/dice-embeddings#pre-trained-models

is reported. These results corroborate our hypothesis: The link prediction performance substantially vary depending on the selection of algebras. For instance, Keci performs well with $Cl_{0,1}$ and $Cl_{4,3}$ on both datasets, whereas Keci performs poorly with $Cl_{0,0}$ and $Cl_{3,0}$. Figure 1 also shows that $Cl_{0,0}$, $Cl_{0,1}$, $Cl_{3,4}$ greatly suffer from overfitting, whereas this is not observed for $Cl_{4,3}$.

Fig. 1. MRR performance of Keci with different p and q for $Cl_{p,q}$.



Depending on the selection of $Cl_{p,q}$, Keci can greatly benefit from the early stopping technique on UMLS, e.g., terminating the training after observing consecutive decreases in the validation performance [26]. For instance, although Keci with $Cl_{0,1}$ reaches its peak generalization performance around 50-75 epochs, training longer decreases its generalization performance. Yet, a possible overfitting is not observed for any configuration on KINSHIP.

6.2 Comparison with other approaches

Tables 3 to 5 report the link prediction results on WN18RR, FB15K-237, YAGO3-10, NELL-995-h25, NELL-995-h50, and NELL-995-h75, respectively. We report the link prediction performance of models on the training, validation, and test splits of the respective dataset to allow for a fine-grained performance analysis, e.g., by allowing for overfitting/underfitting to be detected. Keci is trained with the dimension scaling technique (elucidated in Section 4.3). Overall, our results corroborates our hypothesis: the suitability of algebras is contingent upon the dataset that is to be embedded. For instance, although DistMult reaches the second best performance on FB15K-237, YAGO3-10, and WN18RR, DistMult performs poorly on NELL-995-h25, UMLS, and KINSHIP. Similarly, although QMult outperforms DistMult, ComplEx, and OMult on FB15K-237, QMult performs poorly on NELL-995-h25, NELL-995-h50, and KINSHIP. Note that all models have the same number of parameters, i.e., DistMult, ComplEx, QMult, OMult, and Keci operate in $\mathbb{R}^d, \mathbb{C}^{d/2}, \mathbb{H}^{d/4}\ \mathbb{O}^{d/4}$, and $Cl_{p,q}(\mathbb{R}^m)$. Note that each baseline model applies an element-wise vector multiplication followed by an inner product in a respectively fixed algebra, whereas Keci begins the search in $\mathbb{R} = Cl_{0,0}$ and updates the coefficients of the $p+q$ base vectors based on the the training loss further as elucidated in Section 4.3.

Table 3. Link prediction results on FB15K-237, YAGO3-10 and WN18RR. All models have the same number of parameters. Each entity and relation is represented with 32-dimensional real valued vector. Each sequence of three rows for a model report the model performance on the training, validation and test datasets. Bold and underlined results indicate the best results and second best results.

Models	**FB15K-237**				**YAGO3-10**				**WN18RR**			
	MRR	@1	@3	@10	MRR	@1	@3	@10	MRR	@1	@3	@10
DistMult	0.365	0.259	0.412	0.573	0.644	0.564	0.691	0.794	0.932	0.885	0.978	0.993
	0.212	0.140	0.236	0.355	0.252	0.182	0.275	0.385	0.353	0.342	0.357	0.372
	<u>0.213</u>	<u>0.141</u>	<u>0.235</u>	<u>0.351</u>	<u>0.247</u>	<u>0.174</u>	<u>0.278</u>	<u>0.382</u>	<u>0.351</u>	<u>0.340</u>	<u>0.353</u>	<u>0.371</u>
ComplEx	0.336	0.237	0.379	0.534	0.623	0.543	0.669	0.773	0.906	0.879	0.927	0.948
	0.196	0.128	0.214	0.332	0.227	0.158	0.249	0.364	0.308	0.277	0.326	0.359
	0.197	0.129	0.218	0.333	0.230	0.156	0.257	0.373	0.313	0.282	0.331	0.364
QMult	0.338	0.238	0.381	0.537	0.471	0.382	0.516	0.642	0.996	0.996	0.996	0.997
	0.210	0.143	0.229	0.343	0.176	0.113	0.195	0.300	0.313	0.278	0.337	0.366
	0.207	0.139	0.226	0.341	0.179	0.112	0.202	0.309	0.308	0.274	0.332	0.361
OMult	0.323	0.226	0.362	0.517	0.429	0.334	0.479	0.610	0.977	0.971	0.982	0.988
	0.195	0.131	0.210	0.327	0.160	0.099	0.177	0.282	0.298	0.269	0.314	0.353
	0.192	0.127	0.206	0.325	0.163	0.100	0.181	0.288	0.295	0.263	0.314	0.353
Keci	0.496	0.390	0.551	0.699	0.664	0.579	0.718	0.821	0.967	0.952	0.982	0.989
	0.268	0.191	0.291	0.421	0.260	0.180	0.293	0.414	0.357	0.343	0.365	0.379
	0.262	**0.185**	**0.286**	**0.419**	**0.265**	**0.187**	**0.295**	**0.414**	**0.354**	**0.341**	**0.359**	**0.377**

Table 3 show that Keci finds a suitable subspace of $Cl_{p,q}$ that leads to better training, validation and test performances across datasets. Keci outperforms all models in all metrics on FB15K-237 and YAGO3-10. Although Keci starts the training process as DistMult, Keci finds a subspace of $Cl_{p,q}$ that fits the training data better. For instance, Keci outperforms DistMult by 13.1% absolute difference in MRR on the training split of FB15K-237. Surprisingly, although

Keci and QMult reach similar performances on the training split of WN18RR, Keci generalizes better than QMult on WN18RR. This may indicate that learning coefficients for each imaginary dimension of $Cl_{p,q}$ acts as a regularizer. We also observe that as the size of the underlying algebras grows, the performance of a respective model decreases on FB15K-237 and YAGO3-10, e.g., DistMult and ComplEx outperform QMult and OMult in all metrics on FB15K-237 and YAGO3-10. Hence, as the size of the algebra grows, increasing the embedding size d may be beneficial depending on the input dataset Similarly observation is also reported in [13]. Table 3 also show that all models greatly suffer from overfitting on all datasets, particularly, on WN18RR. This highlights the importance of applying regularization.

Tables 4 and 5 show that Keci generalizes better than all baselines on NELL, UMLS, and KINSHIP benchmark datasets. Keci outperforms all baselines models on NELL-005-h25 in all metrics, while baselines reach better training performance on the other NELL datasets.

Table 4. Link prediction results on NELL-995-h25, NELL-995-h50, and NELL-995-h75. All models have the same number of parameters. Each entity and relation is represented with 32-dimensional real valued vector. Each three rows for a model report performance on the training, validation and test datasets. Bold and underlined results indicate the best results and second best results.

Models	**NELL-995-h25**				**NELL-995-h50**				**NELL-995-h75**			
	MRR	@1	@3	@10	MRR	@1	@3	@10	MRR	@1	@3	@10
DistMult	0.683	0.614	0.725	0.808	0.890	0.840	0.930	0.972	0.929	0.893	0.958	0.984
	0.144	0.101	0.157	0.227	0.170	0.117	0.191	0.275	0.166	0.115	0.180	0.272
	0.140	0.097	0.155	0.224	0.178	0.124	0.197	0.282	0.164	0.112	0.179	0.266
ComplEx	0.854	0.804	0.890	0.939	0.968	0.951	0.982	0.992	0.820	0.750	0.877	0.936
	0.165	0.113	0.182	0.265	0.142	0.090	0.157	0.246	0.138	0.093	0.153	0.223
	0.163	0.112	0.180	0.266	0.150	0.098	0.165	0.252	0.135	0.094	0.146	0.217
QMult	0.518	0.450	0.555	0.641	0.667	0.580	0.729	0.823	0.943	0.914	0.972	0.987
	0.113	0.079	0.123	0.179	0.118	0.075	0.134	0.198	0.145	0.094	0.158	0.249
	0.113	0.076	0.125	0.181	0.125	0.081	0.140	0.208	0.152	0.103	0.165	0.246
OMult	0.513	0.446	0.548	0.638	0.710	0.630	0.761	0.859	0.663	0.565	0.736	0.832
	0.109	0.072	0.119	0.181	0.155	0.102	0.170	0.262	0.109	0.071	0.118	0.179
	0.110	0.075	0.121	0.178	0.161	0.107	0.179	0.266	0.110	0.073	0.120	0.177
Keci	0.882	0.831	0.926	0.963	0.587	0.493	0.648	0.758	0.760	0.674	0.823	0.909
	0.207	0.152	0.229	0.314	0.227	0.162	0.256	0.354	0.225	0.158	0.253	0.356
	0.205	**0.152**	**0.224**	**0.310**	**0.227**	**0.161**	**0.254**	**0.355**	**0.216**	**0.152**	**0.242**	**0.341**

Table 5 shows that Keci outperforms all baselines in all metrics. ComplEx and QMult outperform Keci on the training split of UMLS in 4 metrics, whereas Keci outperform them considerably (up to absolute 15% in MRR). Importantly, although DistMult and Keci reaches similar performance in the training split

Table 5. Link prediction results on UMLS and KINSHIP. All models have the same number of parameters. Each entity and relation is represented with 32-dimensional real valued vector. Each three rows per model report performance on the training, validation and test datasets, respectively. Bold and underlined results indicate the best results and second best results.

Models	**UMLS**				**KINSHIP**			
	MRR	@1	@3	@10	MRR	@1	@3	@10
DistMult	0.924	0.887	0.950	0.993	0.657	0.525	0.734	0.938
	0.607	0.471	0.679	0.883	0.511	0.349	0.589	0.874
	0.605	0.469	0.673	0.896	0.520	0.357	0.601	0.888
ComplEx	0.996	0.992	1.000	1.000	0.887	0.818	0.952	0.991
	0.686	0.536	0.801	0.941	0.751	0.627	0.851	0.962
	0.702	0.557	0.813	0.942	<u>0.738</u>	0.614	<u>0.834</u>	<u>0.964</u>
QMult	0.994	0.990	0.999	1.000	0.854	0.772	0.926	0.985
	0.716	0.596	0.807	0.938	0.712	0.575	0.815	0.949
	<u>0.722</u>	<u>0.590</u>	0.816	<u>0.952</u>	0.726	0.597	0.823	0.958
OMult	0.988	0.977	0.999	1.000	0.765	0.658	0.846	0.955
	0.716	0.587	0.810	0.942	0.626	0.481	0.726	0.917
	<u>0.722</u>	0.585	<u>0.836</u>	<u>0.952</u>	0.641	0.497	0.738	0.921
Keci	0.940	0.900	0.976	0.993	0.887	0.823	0.943	0.988
	0.854	0.775	0.919	0.973	0.768	0.648	0.867	0.970
	0.850	**0.768**	**0.917**	**0.976**	**0.764**	**0.644**	**0.855**	**0.974**

in terms of MRR performance (e.g., 2% absolute difference in MRR), Keci generalizes considerably better than DisMult (e.g., circa 25% absolute difference in MRR). This is an important result as it implies that although Keci starts the search in $\mathbb{R}$ as DistMult does, Keci finds coefficients for base vectors that leads to an improvement in the generalization. We observe that although ComplEx, QMult and OMult reach on-par link prediction performance on the training dataset of UMLS, this observation cannot be made on KINSHIP. OMult performs worse than ComplEx, QMult on the training, validation and test splits of KINSHIP. This again corroborates our hypothesis that the selection of the algebra within which a knowledge graph embedding model operates has a tangible impact in the link prediction performance.

7 Conclusion

We introduced the first knowledge graph embedding model–Keci–that can parameterize the algebra within which embeddings for entities and relation are learned. With Keci, the selection of an underlying algebra can be performed in a data-driven fashion. Our extensive experiments on seven benchmark datasets suggest that this ability leads Keci to outperform state-of-the-art models in all metrics. Importantly, our results also show that Learning to scale embedding dimensions makes Keci more robust against overfitting.

Acknowledgements This work has been supported by the HORIZON Europe research and innovation programme (GA No 101070305), by the Ministry of Culture and Science of North Rhine-Westphalia (GA No NW21-059D), by the German Research Foundation (GA No TRR 318/1 2021 – 438445824) and by the H2020 Marie Skłodowska-Curie programme (GA No 860801).

References

1. Balažević, I., Allen, C., Hospedales, T.M.: Hypernetwork knowledge graph embeddings. In: Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings 28. pp. 553–565. Springer (2019)
2. Balažević, I., Allen, C., Hospedales, T.M.: Tucker: Tensor factorization for knowledge graph completion. arXiv preprint arXiv:1901.09590 (2019)
3. Bonner, S., Barrett, I.P., Ye, C., Swiers, R., Engkvist, O., Hoyt, C.T., Hamilton, W.L.: Understanding the performance of knowledge graph embeddings in drug discovery. Artificial Intelligence in the Life Sciences **2**, 100036 (2022)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems **26** (2013)
5. Brandstetter, J., Berg, R.v.d., Welling, M., Gupta, J.K.: Clifford neural layers for pde modeling. arXiv preprint arXiv:2209.04934 (2022)
6. Chami, I., Wolf, A., Juan, D.C., Sala, F., Ravi, S., Ré, C.: Low-dimensional hyperbolic knowledge graph embeddings. arXiv preprint arXiv:2005.00545 (2020)
7. Dai, Y., Wang, S., Xiong, N.N., Guo, W.: A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics **9**(5), 750 (2020)
8. Demir, C., Moussallem, D., Heindorf, S., Ngomo, A.C.N.: Convolutional hypercomplex embeddings for link prediction. In: Asian Conference on Machine Learning. pp. 656–671. PMLR (2021)
9. Demir, C., Ngomo, A.C.N.: Convolutional complex knowledge graph embeddings. In: The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18. pp. 409–424. Springer (2021)
10. Demir, C., Ngomo, A.C.N.: Hardware-agnostic computation for large-scale knowledge graph embeddings. Software Impacts **13**, 100377 (2022)
11. Demir, C., Ngomo, A.C.N.: Learning permutation-invariant embeddings for description logic concepts. arXiv preprint arXiv:2303.01844 (2023)
12. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
13. Gregucci, C., Nayyeri, M., Hernández, D., Staab, S.: Link prediction with attention applied on multiple knowledge graph embedding models. arXiv preprint arXiv:2302.06229 (2023)
14. Hamilton, W., Bajaj, P., Zitnik, M., Jurafsky, D., Leskovec, J.: Embedding logical queries on knowledge graphs. Advances in neural information processing systems **31** (2018)

15. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
16. Hitzer, E.: Extending lasenby's embedding of octonions in space-time algebra c l (1, 3) cl\left (1, 3\right), to all three-and four dimensional clifford geometric algebras c l (p, q), n= p+ q= 3, 4 cl\left (p, q\right), n= p+ q= 3, 4. Mathematical Methods in the Applied Sciences (2022)
17. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.d., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. ACM Computing Surveys (CSUR) **54**(4), 1–37 (2021)
18. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers). pp. 687–696 (2015)
19. Lacroix, T., Usunier, N., Obozinski, G.: Canonical tensor decomposition for knowledge base completion. In: International Conference on Machine Learning. pp. 2863–2872. PMLR (2018)
20. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29 (2015)
21. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: 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, Volume 2 (Short Papers). pp. 327–333. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-2053, https://aclanthology.org/N18-2053
22. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proceedings of the IEEE **104**(1), 11–33 (2015)
23. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 30 (2016)
24. Nickel, M., Tresp, V., Kriegel, H.P., et al.: A three-way model for collective learning on multi-relational data. In: Icml. vol. 11, pp. 3104482–3104584 (2011)
25. Paliwal, S., de Giorgio, A., Neil, D., Michel, J.B., Lacoste, A.: Preclinical validation of therapeutic targets predicted by tensor factorization on heterogeneous graphs. Scientific reports **10**(1), 1–19 (2020)
26. Prechelt, L.: Early stopping—but when? Neural networks: tricks of the trade: second edition pp. 53–67 (2012)
27. Ren, F., Li, J., Zhang, H., Liu, S., Li, B., Ming, R., Bai, Y.: Knowledge graph embedding with atrous convolution and residual learning. arXiv preprint arXiv:2010.12121 (2020)
28. Ruffinelli, D., Broscheit, S., Gemulla, R.: You CAN teach an old dog new tricks! on training knowledge graph embeddings. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=BkxSmlBFvr
29. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
30. Trouillon, T., Dance, C.R., Gaussier, E., Welbl, J., Riedel, S., Bouchard, G.: Knowledge graph completion via complex tensor factorization. J. Mach. Learn. Res. **18**(1), 4735–4772 (jan 2017)

31. Trouillon, T., Dance, C.R., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Knowledge graph completion via complex tensor factorization. arXiv preprint arXiv:1702.06879 (2017)
32. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International conference on machine learning. pp. 2071–2080. PMLR (2016)
33. Wang, M., Qiu, L., Wang, X.: A survey on knowledge graph embeddings for link prediction. Symmetry **13**(3), 485 (2021)
34. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
35. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI conference on artificial intelligence. vol. 28 (2014)
36. Xiong, W., Hoang, T., Wang, W.Y.: Deeppath: A reinforcement learning method for knowledge graph reasoning. arXiv preprint arXiv:1707.06690 (2017)
37. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
38. Ye, Z., Kumar, Y.J., Sing, G.O., Song, F., Wang, J.: A comprehensive survey of graph neural networks for knowledge graphs. IEEE Access **10**, 75729–75741 (2022)
39. Yi, H.C., You, Z.H., Huang, D.S., Kwoh, C.K.: Graph representation learning in bioinformatics: trends, methods and applications. Briefings in Bioinformatics **23**(1), bbab340 (2022)
40. Yu, M., Bai, C., Yu, J., Zhao, M., Xu, T., Liu, H., Li, X., Yu, R.: Translation-based embeddings with octonion for knowledge graph completion. Applied Sciences **12**(8), 3935 (2022)
41. Zamini, M., Reza, H., Rabiei, M.: A review of knowledge graph completion. Information **13**(8), 396 (2022)
42. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. Advances in neural information processing systems **32** (2019)