Knowledge Graph Embeddings in Geometric Algebras

Chengjin Xu
University of Bonn / Germany
xuc@iai.uni-bonn.de

Mojtaba Nayyeri
University of Bonn / Germany
nayyeri@iai.uni-bonn.de

Yung-Yu Chen
University of Bonn / Germany
s6ynchen@uni-bonn.de

Jens Lehmann
University of Bonn / Germany
Fraunhofer IAIS/ Germany
jens.lehmann@iais.fraunhofer.de

Abstract

Knowledge graph (KG) embedding aims at embedding entities and relations in a KG into a low dimensional latent representation space. Existing KG embedding approaches model entities and relations in a KG by utilizing real-valued , complex-valued, or hypercomplex-valued (Quaternion or Octonion) representations, all of which are subsumed into a geometric algebra. In this work, we introduce a novel geometric algebra-based KG embedding framework, GeomE, which utilizes multivector representations and the geometric product to model entities and relations. Our framework subsumes several state-of-the-art KG embedding approaches and is advantageous with its ability of modeling various key relation patterns, including (anti-)symmetry, inversion and composition, rich expressiveness with higher degree of freedom as well as good generalization capacity. Experimental results on multiple benchmark knowledge graphs show that the proposed approach outperforms existing state-of-the-art models for link prediction.

1 Introduction

Knowledge graphs (KGs) are directed graphs where nodes represent entities and (labeled) edges represent the types of relationships among entities. This can be represented as a collection of triples (h, r, t), each representing a relation r between a "head-entity" h and an "tail-entity" t. Some real-world knowledge graphs include Freebase (Bollacker et al., 2008), WordNet (Miller, 1995), YAGO (Suchanek et al., 2007), and DBpedia (Auer et al., 2007).

However, most existing KGs are incomplete. The task of link prediction alleviates this drawback by inferring missing facts based on the known facts in a KG and thus has gained growing interest. Embedding KGs into a low-dimensional space and learning latent representations of entities and relations in KGs is an effective solution for this task. In general, most existing KG embedding models learn to embed KGs by optimizing a scoring function which assigns higher scores to true facts than invalid ones.

Recently, learning KG embeddings in the complex or hypercomplex spaces has been proven to be a highly effective inductive bias. ComplEx (Trouillon et al., 2016), RotatE, pRotatE (Sun et al., 2019), and QuatE (Zhang et al., 2019) achieved the state-of-the-art results on link prediction, due to their abilities of capturing various relations (i.e., modeling symmetry and anti-symmetry). They both use the asymmetrical Hermitian product to score relational triples where the components of entity/relation embeddings are complex numbers or quaternions.

Complex numbers and quaternions can be described by the various components within a Clifford multivector (Chappell et al., 2015). In other words, the geometric algebra of Clifford (1882) provides an elegant and efficient rotation representation in terms of multivector which is more general than Hamilton (1844)'s unit quaternion.

In this paper, we propose a novel KG embedding approach, GeomE, which is based on Clifford multivectors and the geometric product. Concretely, we utilize N multivector embeddings of N grades ($N = 2, 3$) to represent entity and relation. Each component of an entity/relation embedding is a multivector in a geometric algebra of N grades, $\mathbb{G}^N$, with scalars, vectors and bivectors, as well as trivectors

Proceedings of the 28th International Conference on Computational Linguistics, pages 530–544
Barcelona, Spain (Online), December 8-13, 2020

(for $N = 3$). In terms of a triple (h, r, t), we use an asymmetrical geometric product which involves the conjugation of the embedding of the tail entity to multiply the embeddings of e_s, r, e_o, and obtain the final score of the triple from the product embedding.

The advantages of our formulas include the following points:

- Our framework GeomE subsumes ComplEx, pRotatE and QuatE. A complex number can be regarded as a scalar plus a bivector in the geometric algebra $\mathbb{G}^2$. A quaternion is isomorphic with a scalar plus three bivectors in the geometric algebra $\mathbb{G}^3$. Thus, GeomE inherits the excellent properties of pRotatE, ComplEx and QuatE and has the ability to model various relation patterns, e.g., (anti-)symmetry, inversion and composition.
- The geometric product units the Grassmann (1844) and Hamilton (1844) algebras into a single structure. Compared to the Hamilton operator used in QuatE, the geometric product provides a greater extent of expressiveness since it involves the operator for vectors, trivectors and n-vectors, in addition to scalars and bivectors.
- Our proposed approach GeomE is not just a single KG embedding model. GeomE can be generalized in the geometric algebras of different grades and is hence more flexible in the expressiveness compared to pRotatE, ComplEx and QuatE. In this paper, we propose two new KG embedding models, i.e., GeomE2D and GeomE3D, based on multivectors from $\mathbb{G}^2$ and $\mathbb{G}^3$, and test their combination model GeomE+.

Experimental results demonstrate that our approach achieves state-of-the-art results on four well-known KG benchmarks, i.e., WN18, FB15K, WN18RR, and FB15K-237.

2 Related Work

Most KG embedding models can be classified as distance-based or semantic matching based, according to their scoring functions.

Distance-based scoring functions aim to learn embeddings by representing relations as translations from head entities to tail entities. Bordes et al. (2013) proposed TransE by assuming that the added embedding of s and r should be close to the embedding of o. Since that, many variants and extensions of TransE have been proposed. For example, TransH (Wang et al., 2014) projects entities and relations into a hyperplane. TransR (Lin et al., 2015) introduces separate projection spaces for entities and relations. TransD (Ji et al., 2015) uses independent projection vectors for each entity and relation and can reduce the amount of calculation compared to TransR. TorusE (Ebisu and Ichise, 2018) defines embeddings and distance function in a compact Lie group, torus. The recent distance-based KG embedding models, RotatE and pRotatE (Sun et al., 2019), propose a rotation-based distance scoring functions with complex-valued embeddings. Likewise, TransComplEx (Nayyeri et al., 2019) also maps entities and relations into a complex-valued vector space.

On the other hand, semantic matching models include RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), SimplE (Kazemi and Poole, 2018) and QuatE (Zhang et al., 2019). In RESCAL, each relation is represented with a square matrix, while DistMult replaces it with a diagonal matrix in order to reduce the complexity. SimplE is also a simple yet effective bilinear approach for knowledge graph embedding. ComplEx embeds entities and relations in a complex space and utilizes an asymmetric Hermitian product to score triples, which is immensely helpful in modeling various relation patterns. QuatE extends ComplEx in a hypercomplex space and replaces the Hermitian product with the Hamilton product which provides a greater extent of expressiveness. In addition, neural network based KG embedding models have also been proposed, e.g., NTN (Socher et al., 2013), ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2019) and IteractE (Vashishth et al., 2020).

Our proposed approach, GeomE, subsumes ComplEx, pRotatE and QuatE in the geometric algebras. In addition to the inheritance of the attractive properties of these existing KG embedding models, our approach takes advantages of the multivectors, e.g., the rich geometric meanings, the excellent representation ability and the generalization ability in the geometric algebras of different grades. Due to the above merits of the geometric algebras and multivectors, they have also been widely applied in computer vision and neurocomputing (Bayro-Corrochano, 2018).

3 Geometric Algebra and Multivectors

Leaning on the earlier concepts of Grassmann (1844)'s exterior algebra and Hamilton (1844)'s quaternions, Clifford (1882) intended his geometric algebra to describe the geometric properties of scalars, vectors and eventually higher dimensional objects. In addition to the well known scalar and vector elements, there are bivectors, trivectors, n-vectors and multivectors which are higher dimensional generalisations of vectors. An N-dimensional vector space $\mathbb{R}^N$ can be embedded in a geometric algebra of N grades , $\mathbb{G}^N$. In this section, we take $\mathbb{G}^2$ and $\mathbb{G}^3$ as examples to introduce multivectors and some corresponding operators.

3.1 2-Grade and 3-Grade Multivectors

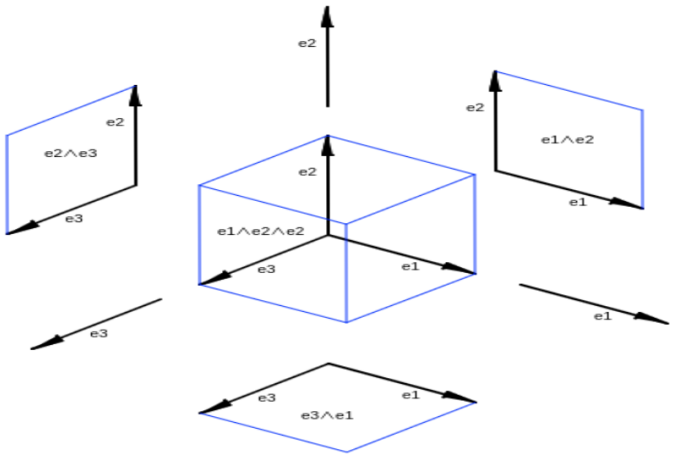

Figure 1: An example of a 3-grade multivector space $\mathbb{G}^3$

Let $\{e_1, e_2, e_3\}$ be an orthonormal basis of $\mathbb{R}^3$. The algebra $\mathbb{G}^3$ is based on two rules: (a) $e_i e_i = 1$,; (b) $e_i e_j = -e_j e_i$, where $i \neq j$. The multivector space $\mathbb{G}^3$ is 8-dimensional with basis:

$$\begin{aligned} 1 &\text{ spans 0-vectors, scalars,} \\ \{e_1, e_2, e_3\} &\text{ spans 1-vectors, vectors,} \\ \{e_1e_2, e_2e_3, e_1e_3\} &\text{ spans 2-vectors, bivectors, and} \\ \{e_1e_2e_3\} &\text{ spans 3-vectors, trivectors.} \end{aligned}$$

Hence an arbitrary 3-grade multivector $M \in \mathbb{G}^3$ can be written as

$$M = a_0 + a_1e_1 + a_2e_2 + a_3e_3 + a_{12}e_1e_2 + a_{23}e_2e_3 + a_{13}e_1e_3 + a_{123}e_1e_2e_3,$$

where $a_0, a_1, a_2, a_3, a_{12}, a_{23}, a_{13}, a_{123}$ are all real numbers. Each element of a multivector, e.g., a scalar, a vector, or an N-vector, is called as a **blade**. A 2-grade multivector $M \in \mathbb{G}^2$ is build from one scalar, two vectors and one bivector.

$$M = a_0 + a_1e_1 + a_2e_2 + a_{12}e_1e_2.$$

The norm of a multivector is equal to the root of the square sum of real values of all blades. Taking the 2-grade multivector as an example, its norm is defined as:

$$||M|| = \sqrt{a_0^2 + a_1^2 + a_2^2 + a_{12}^2}. \tag{1}$$

3.2 Multivectors vs Quaternions

Quaternions are elements of the form: $Q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$, where q_0, q_1, q_2, q_3 are real numbers and **i**, **j**, **k** are three different square roots of -1 and are the new elements used for the construction of quaternions. They have the following algebraic properties: $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$

Bivectors from $\mathbb{G}^3$ have similar algebraic properties as the basis of the quaternion space.

$$\begin{aligned} &(e_ie_j)^2 = -e_ie_je_je_i = -1 \text{ where } i, j = 1, 2, 3, \text{ and } i \neq j \\ &e_1e_2e_2e_3e_1e_3 = e_1e_3e_1e_3 = -1 \end{aligned} \tag{2}$$

Thus we can embed a quaternion in a 3-grade geometric algebra $\mathbb{G}^3$ with a scalar plus three bivectors. A complex number can likewise be regarded as a scalar plus one bivector from $\mathbb{G}^2$.

3.3 Geometric Product and Clifford Conjugation

Geometric algebra also introduces a new product, **geometric product**, as well as three multivector involutions, **space inversion**, **reversion** and **Clifford conjugation**.

The geometric product of two multivectors comprises of multiplications between scalars, bivectors, trivectors and n-vectors. The product of two 2-grade multivectors $M_a = a_0 + a_1e_1 + a_2e_2 + a_{12}e_1e_2$ and $M_b = b_0 + b_1e_1 + b_2e_2 + b_{12}e_1e_2$ from $\mathbb{G}^2$ is equal to

$$\begin{aligned} M_a \otimes_2 M_b =& a_0b_0 + a_1b_1 + a_2b_2 - a_{12}b_{12} + (a_0b_1 + a_1b_0 - a_2b_{12} + a_{12}b_2)e_1 \\ &+ (a_0b_2 + a_1b_{12} + a_2b_0 - a_{12}b_1)e_2 + (a_0b_{12} + a_1b_2 - a_2b_1 + a_{12}b_0)e_1e_2. \end{aligned} \tag{3}$$

The product of two 3-grade multivectors $M_a = a_0 + a_1e_1 + a_2e_2 + a_3e_3 + a_{12}e_1e_2 + a_{23}e_2e_3 + a_{13}e_1e_3 + a_{123}e_1e_2e_3$ and $M_b = b_0 + b_1e_1 + b_2e_2 + b_3e_3 + b_{12}e_1e_2 + b_{23}e_2e_3 + b_{13}e_1e_3 + b_{123}e_1e_2e_3$ from $\mathbb{G}^3$ is represented in Appendix B.

Clifford Conjugation: The Clifford Conjugation of an n-grade multivector M is a subsequent composition of **space inversion** M^* and **reversion** $M^\dagger$ as $\overline{M} = M^{\dagger *}$, where **space inversion** M^* is obtained by changing e_i to $-e_i$ and **reversion** is obtained by reversing the order of all products i.e. changing $e_1e_2\cdots e_n$ to $e_ne_{n-1}\cdots e_1$. For example, the conjugation of $M \in \mathbb{G}^2$, which is formed as $M = A_0 + A_1 + A_2$ with $A_0 = a_0$, $A_1 = a_1e_1 + a_2e_2$, $A_2 = a_{12}e_1e_2$, is computed as $\overline{M} = A_0 - A_1 - A_2$. Note that the product of a multivector M and its conjugation $\overline{M}$ is always a scalar. For a given 2-grade multivector $M = a_0 + a_1e_1 + a_2e_2 + a_{12}e_1e_2$, we have

$$M \otimes_2 \overline{M} = a_0^2 - a_1^2 - a_2^2 + a_{12}^2, \tag{4}$$

producing a real number, though not necessarily non-negative.

4 Our Method

4.1 Knowledge Graph Embedding Model based on Geometric Algebras

Let $\mathcal{E}$ denote the set of all entities and $\mathcal{R}$ the set of all relations present in a knowledge graph. A triple is represented as (h, r, t), with $h, t \in \mathcal{E}$ denoting head and tail entities respectively and $r \in \mathcal{R}$ the relation between them. We use $\Omega = \{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ to denote the set of observed triples. The key issue of KG embeddings is to represent entities and relations in a continuous low-dimensional space.

Our approach GeomE uses the geometric product and multivectors for KG embedding. In this paper, we propose two models built with our approach, GeomE2D and GeomE3D, based on 2-grade multivectors and 3-grade multivectors respectively.

GeomE2D represents each entity/relation as a k dimensional embedding $\mathbf{M}$ where each element is a 2-grade multivector, i.e., $\mathbf{M} = [M_1, \ldots, M_k]$, $M_i \in \mathbb{G}^2, i = 1, \ldots, k$, where k is the dimensionality of embeddings. Given a triple (h, r, t), we represent embeddings of h, r and t by $\mathbf{M}_h = [M_{h_1}, \ldots, M_{h_k}], \mathbf{M}_r = [M_{r_1}, \ldots, M_{r_k}]$, and $\mathbf{M}_t = [M_{t_1}, \ldots, M_{t_k}]$ respectively. Note that each element of $\mathbf{M}$ is a 2-grade multivector. For example, $M_{h_1} = \{h_0^1 + h_1^1e_1 + h_2^1e_2 + h_{12}^1e_1e_2,\ h_0^1, h_1^1, h_2^1, h_{12}^1 \in \mathbb{R}\}$.

GeomE3D embeds h, r, t into k dimensional embeddings $\mathbf{M}_h, \mathbf{M}_r$ and $\mathbf{M}_t$ respectively where each element of the embeddings is a 3-grade multivector i.e. $M_{h_i}, M_{r_i}, M_{t_i} \in \mathbb{G}^3$ for $i = 1, \ldots, k$, where $M_{h_i} = h_0^i + h_1^ie_1 + h_2^ie_2 + h_3^ie_3 + h_{12}^ie_1e_2 + h_{23}^ie_2e_3 + h_{13}^ie_1e_3 + h_{123}^ie_1e_2e_3$.

Scoring Function of GeomE is defined as the scalar of the product of the embeddings of h, r and t by using the geometric product and the Clifford conjugation.

$$\phi^{GeomE}(h, r, t) = \langle \mathbf{Sc}(\mathbf{M}_h \otimes_n \mathbf{M}_r \otimes_n \overline{\mathbf{M}_t}), \mathbf{1} \rangle, \tag{5}$$

where $n = 2$ for GeomE2D and $n = 3$ for GeomE3D, $\otimes_n$ denotes element-wise **Geometric Product** between two k dimensional n-grade multivectors (e.g. $\mathbf{M}_h \otimes_n \mathbf{M}_r = [M_{h_1} \otimes_n M_{r_1}, \cdots, M_{h_k} \otimes_n M_{r_k}]$), **Sc** denotes the scalar component of a multivector, $\mathbf{1}$ denotes a $k \times 1$ vector having all k elements equal to one, $\overline{\mathbf{M}}$ denotes the element-wise conjugation of multivectors i.e. $\overline{\mathbf{M}} = [\overline{M}_1, \ldots, \overline{M}_k]$. The expanded formulation of scoring functions for GeomE2D and GeormE3D are presented in the Appendix C.

4.2 Training

Most of previous semantic matching models, e.g., ComplEx, are learned by minimizing a sampled binary logistic loss function (Trouillon et al., 2016). Motivated by the solid results in (Lacroix et al., 2018), we formulate the link prediction task as a multiclass classification problem by using a full multiclass log-softmax loss function, and apply N3 regularization and reciprocal approaches for our models.

Given a training set $\Omega \ni (h, r, t)$, we create a reciprocal training set $\Omega^* \ni (t, r^{-1}, h)$ by adding reverse relations and the instantaneous multiclass log-softmax loss is defined as:

$$\begin{aligned}\mathcal{L} = \sum_{(h,r,t)\in(\Omega\cup\Omega^*)} [&- \log(\frac{\exp(\phi(h,r,t))}{\sum_{h'\in\mathcal{E}}\exp(\phi(h',r,t))}) - \log(\frac{\exp(\phi(h,r,t))}{\sum_{t'\in\mathcal{E}}\exp(\phi(h,r,t'))}) \\ &+ \frac{\lambda}{3}\sum\nolimits_{i=1}^{k}(||M_{h_i}||_3^3 + ||M_{r_i}||_3^3 + ||M_{t_i}||_3^3)]\end{aligned} \tag{6}$$

N3 regularization and reciprocal learning approaches have been proven to be helpful in boosting the performances of semantic matching models (Lacroix et al., 2018; Zhang et al., 2019). Different from the sampled binary logistic loss function which generates a certain number of negative samples for each training triple by randomly corrupting the head or tail entity, the full multiclass log-softmax considers all possible negative samples and thus has a fast converge speed. On FB15K, the training process of a GeomE3D model with a high dimensionality of $k = 1000$ needs less than 100 epochs and cost about 4 minutes per epoch on a single GeForce RTX 2080 device.

4.3 Connection to QuatE, Complex and pRotatE

As mentioned in Section 3, a bivector unit in a geometric algebra has similar properties to an imaginary unit in a complex or hypercomplex space. Thus, a quaternion is isomorphic with a 3-grade multivector consisting of a scalar and three bivectors, and a complex value can be regarded as a 2-grade multivector consisting of a scalar and one bivector.

Subsumption of QuatE: By setting the coefficients of vectors and trivectors of $\mathbf{M}_h$, $\mathbf{M}_r$, and $\mathbf{M}_t$ in Equation 5 to zero, we obtain the following equations for GeomE3D

$$\begin{aligned}&\phi^{GeomE3D}(h,r,t) = \\ &(\mathbf{h}_0\circ\mathbf{r}_0 - \mathbf{h}_{12}\circ\mathbf{r}_{12} - \mathbf{h}_{23}\circ\mathbf{r}_{23} - \mathbf{h}_{13}\circ\mathbf{r}_{13})\circ\mathbf{t}_0 + (\mathbf{h}_0\circ\mathbf{r}_{12} + \mathbf{h}_{12}\circ\mathbf{r}_0 - \mathbf{h}_{13}\circ\mathbf{r}_{23} + \mathbf{h}_{23}\circ\mathbf{r}_{13})\circ\mathbf{t}_{12} + \\ &(\mathbf{h}_0\circ\mathbf{r}_{23} + \mathbf{h}_{23}\circ\mathbf{r}_0 - \mathbf{h}_{12}\circ\mathbf{r}_{13} + \mathbf{h}_{13}\circ\mathbf{r}_{12})\circ\mathbf{t}_{23} + (\mathbf{h}_0\circ\mathbf{r}_{13} + \mathbf{h}_{13}\circ\mathbf{r}_0 + \mathbf{h}_{12}\circ\mathbf{r}_{23} - \mathbf{h}_{23}\circ\mathbf{r}_{12})\circ\mathbf{t}_{13}.\end{aligned} \tag{7}$$

where $\circ$ denotes Hadamard product, $\mathbf{h}_j = [h_j^1, \ldots, h_j^k],\ j \in \{0, 12, 23, 13\}$. We can find that Equation 7 recovers the form of the scoring function of QuatE regardless of the normalization of the relational quaternion. Therefore, GeomE3D *subsumes* the QuatE model.

Subsumption of ComplEx: By setting the coefficient of vectors $\mathbf{M}_h$, $\mathbf{M}_r$, and $\mathbf{M}_t$ to zero in Equation 5 for GeomE2D, we obtain

$$\phi^{GeomE2D}(h,r,t) = (\mathbf{h}_0\circ\mathbf{r}_0 - \mathbf{h}_{12}\circ\mathbf{r}_{12})\circ\mathbf{t}_0 + (\mathbf{h}_0\circ\mathbf{r}_{12} + \mathbf{h}_{12}\circ\mathbf{r}_0)\circ\mathbf{t}_{12}, \tag{8}$$

where $\mathbf{h}_j = [h_j^1, \ldots, h_j^k],\ j \in \{0, 12\}$. The Equation 8 recovers the form of the scoring function of ComplEx. Therefore, GeomE2D *subsumes* the ComplEx model. Additinally, comparing equations 7 and 8, we conclude that GeomE3D also *subsumes* ComplEx.

Subsumption of pRotatE: Apart from ComplEx and QuatE, GeomE also subsumes pRotatE. We start from the formulation of the scoring function of pRotatE and show that the scoring function is a special case of Equation 8. The scoring function of pRotatE is defined as

$$\phi^{pRotatE}(h,r,t) = -\|\mathbf{h}\circ\mathbf{r} - \mathbf{t}\|, \tag{9}$$

where the modulus of each element of relation vectors is $|\mathbf{r}_i| = 1, i = 1, \ldots, k$, and $|\mathbf{h}_i| = |\mathbf{t}_i| = C \in \mathbb{R}^+$. After some derivation on the score of pRotatE and GeomE2D (see details in Appendix D), we can obtain $\phi^{GeomE2D}(h,r,t) = \frac{\phi^{pRotatE^2}(h,r,t) - 2kC^2}{2}$. Note that $2kC^2$ is a constant number as k and C, and thus does not affect the overall ranking obtained by computing and sorting the scores of triples.

For a triple (h, r, t), there is a positive correlation between its GeomE score and pRotatE score since pRotatE scores are always non-positive. Therefore, GoemE2D and consequently GeomE3D *subsumes* the pRotatE model in the terms of ranking.

Overall, it can be seen that our framework subsumes ComplEx, pRotatE and QuatE and provides more degrees of freedom by introducing vectors and trivectors. In addition, our framework can be generalized into the geometric algebras with higher grades ($\mathbb{G}^n, n > 3$) and is hence more flexible in expressiveness.

Although we introduce more coefficients in our framework, our models have the same time complexity as pRotatE, ComplEx and QuatE as shown in Table 1. And the memory sizes of our models increase linearly with the dimensionality of embeddings.

Model	Scoreing Function	Relation Parameters	$\mathcal{O}_{time}$	$\mathcal{O}_{space}$
TransE	$-\|\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|\|$	$\mathbf{r} \in \mathbb{R}^k$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
DistMult	$< \mathbf{h}, \mathbf{r}, \mathbf{t} >$	$\mathbf{r} \in \mathbb{R}^k$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
ComplEx	$\mathbf{Re}(< \mathbf{h}, \mathbf{r}, \bar{\mathbf{t}} >)$	$\mathbf{r} \in \mathbb{C}^k$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
ConvE	$f(\text{vec}(f([\mathbf{W_h}; \mathbf{W_r}] * \boldsymbol{\omega}))\mathbf{W})\mathbf{t}$	$\mathbf{W_r} \in \mathbb{R}^{k_w \times k_h}$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
(p)RotatE	$-\|\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|\|$	$\mathbf{r} \in \mathbb{C}^k$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
QuatE	$\mathbf{Q_h} \otimes \mathbf{W_r^{\triangleleft}} \cdot \mathbf{Q_t}$	$\mathbf{W_r} \in \mathbb{H}^k$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
GeomE2D	$\langle \mathbf{Sc}(\mathbf{M}_h \otimes_2 \mathbf{M}_r \otimes_2 \overline{\mathbf{M}_t}), \mathbf{1} \rangle$	$\mathbf{M_r} \in \mathbb{G}^{2\times k}$	$\mathcal{O}(k)$	$\mathcal{O}(k)$
GeomE3D	$\langle \mathbf{Sc}(\mathbf{M}_h \otimes_3 \mathbf{M}_r \otimes_3 \overline{\mathbf{M}_t}), \mathbf{1} \rangle$	$\mathbf{M_r} \in \mathbb{G}^{3\times k}$	$\mathcal{O}(k)$	$\mathcal{O}(k)$

Table 1: Scoring functions of state-of-the-art link prediction models, their parameters as well as their time complexity and space complexity. vec() denotes the matrix flattening. $*$ denotes the convolution operator. f denotes a non-linear function. $\otimes$ denotes the Hamilton product. $\mathbb{H}$ denotes a hypercomplex space. $\triangleleft$ denotes the normalization of quaternions.

4.4 Ability of Modeling Various Relation Patterns

Our framework subsumes pRotatE, ComplEx and QuatE, and thus inherits their attractive properties: One of the merits of our framework is the ability of modeling various patterns including **symmetry/anti-symmetry, inverse** and **composition**. We give the formal definitions of these relation patterns.

Definition *A relation r is **symmetric (anti-symmetric)**, if* $\forall h, t \;\; r(h,t) \Rightarrow r(t,h)$ $(r(h,t) \Rightarrow \neg r(t,h))$.
Definition *A relation* r_1 *is **inverse** of relation* r_2*, if* $\forall h, t \;\; r_1(h,t) \Rightarrow r_2(t,h)$.
Definition *relation* r_3 *is **composed of** relation* r_1, r_2*, if* $\forall h, o, t \;\; r_1(h,o) \wedge r_2(o,t) \Rightarrow r_3(h,t)$.
*Clauses with the above-mentioned forms are **(anti-)symmetry, inversion** and **composition** patterns.*

GeomE can infer and model various relation patterns defined above by taking advantages of the flexibility and representational power of geometric algebras and the geometric product.

(Anti-)symmetry: By utilizing the conjugation of embeddings of tail entities, our framework can model (anti-)symmetry patterns. The symmetry property of GeomE2D and GeomE3D can be proved by enforcing the coefficients of vectors and bivectors in relation embeddings to be zero. On the other hand, their scoring functions are asymmetric about relations when the coefficients of vectors and bivetors in relation embeddings are nonzero. For GeomE2D, the difference score of (h, r, t) and (t, r, h) is equal to

$$\begin{aligned}\phi^{GeomE2D}(h,r,t) - \phi^{GeomE2D}(t,r,h) = 2[(\mathbf{h}_1 \circ \mathbf{t}_0 - \mathbf{h}_0 \circ \mathbf{t}_1 + \mathbf{h}_{12} \circ \mathbf{t}_2 - \mathbf{h}_2 \circ \mathbf{t}_{12}) \circ \mathbf{r}_1 + \\ (\mathbf{h}_2 \circ \mathbf{t}_0 - \mathbf{h}_0 \circ \mathbf{t}_2 + \mathbf{h}_1 \circ \mathbf{t}_{12} - \mathbf{h}_{12} \circ \mathbf{t}_1) \circ \mathbf{r}_2 + (\mathbf{h}_0 \circ \mathbf{t}_{12} - \mathbf{h}_{12} \circ \mathbf{t}_0 + \mathbf{h}_2 \circ \mathbf{t}_1 - \mathbf{h}_1 \circ \mathbf{t}_2) \circ \mathbf{r}_{12}].\end{aligned} \tag{10}$$

This difference is equal to zero when $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_{12} = \mathbf{0}$. Embeddings of multiple symmetric relations could still express their different semantics since their scalar parts might be different.

Inversion: As for a pair of inverse relations r and r', the scores of (h, r, t) and (t, r', h) are equal when $\mathbf{M}_r = \overline{\mathbf{M}_{r'}}$. Concretely, the difference score of (h, r, t) and (t, r', h) is equal to

$$\begin{aligned}\phi^{GeomE2D}(h,r,t) - \phi^{GeomE2D}(t,r',h) = (\mathbf{h}_1 \circ \mathbf{t}_0 - \mathbf{h}_0 \circ \mathbf{t}_1 + \mathbf{h}_{12} \circ \mathbf{t}_2 - \mathbf{h}_2 \circ \mathbf{t}_{12}) \circ (\mathbf{r}_1 + \mathbf{r'}_1) + \\ (\mathbf{h}_2 \circ \mathbf{t}_0 - \mathbf{h}_0 \circ \mathbf{t}_2 + \mathbf{h}_1 \circ \mathbf{t}_{12} - \mathbf{h}_{12} \circ \mathbf{t}_1) \circ (\mathbf{r}_2 + \mathbf{r'}_2) + (\mathbf{h}_0 \circ \mathbf{t}_{12} - \mathbf{h}_{12} \circ \mathbf{t}_0 + \mathbf{h}_2 \circ \mathbf{t}_1 - \mathbf{h}_1 \circ \mathbf{t}_2) \\ \circ (\mathbf{r}_{12} + \mathbf{r'}_{12}) + (\mathbf{h}_0 \circ \mathbf{t}_0 - \mathbf{h}_1 \circ \mathbf{t}_1 - \mathbf{h}_2 \circ \mathbf{t}_2 + \mathbf{h}_{12} \circ \mathbf{t}_{12}) \circ (\mathbf{r}_0 - \mathbf{r'}_0).\end{aligned} \tag{11}$$

This difference is equal to zero when $\mathbf{r}_0 = \mathbf{r'}_0, \mathbf{r}_1 = -\mathbf{r'}_1, \mathbf{r}_2 = -\mathbf{r'}_2, \mathbf{r}_{12} = -\mathbf{r'}_{12}$.

Composition: GeomE can also model composition patterns by introducing some constraints on embeddings. The detailed proof can be found in Appendix E.

5 Experiments and Results

5.1 Experimental Setup

Datasets We use four widely used KG benchmarks for evaluating our proposed models, i.e., FB15K, WN18, FB15K-237 and WN18RR. The statics of these datasets are listed in Table 6. FB15K and WN18 are introduced in (Bordes et al., 2013). The former is extracted from FreeBase (Bollacker et al., 2008), and the latter is a subsampling of WordNet (Miller, 1995). It is firstly discussed in (Toutanova et al., 2015) that WN18 and FB15K suffer from test leakage through inverse relations, i.e. many test triples can be obtained simply by inverting triples in the training set. To address this issue, Toutanova et al. (2015) generated FB15K-237 by removing inverse relations in FB15K. Likewise, Dettmers et al (Dettmers et al., 2018) generated WN18RR by removing inverse relations in WN18. The recent literature shows that FB15K-237 and WN18RR are harder to fit and thus more challenging for new KG embedding models. The details of dataset statistics are listed in the Appendix F.

Evaluation Protocols Link prediction is to complete a fact with a missing entity. Given a test triple (h, r, t), we corrupt this triple by replacing h or t with all possible entities, sort all the corrupted triples based on their scores and compute the rank of the test triple. Three evaluation metrics are used here, **Mean Rank (MR)**, **Mean Reciprocal Rank (MRR)** and **Hits@k**. We also apply the filtered setting proposed in (Bordes et al., 2013).

Implementation Details We used Adagrad (Duchi et al., 2011) as the optimizer and fine-tuned the hyperparameters on the validation dataset. We fixed batch size $b = 1000$ and learning rate $lr = 0.1$. We decided to focus on influences of the embedding dimensionality $k \in \{20, 50, 100, 200, 500, 1000\}$ and the regularization coefficient $\lambda \in \{0.0001, 0.00025, 0.0005, 0.00075, 0.001, \cdots, 0.1\}$. The default configuration for our proposed models is as follows: $lr = 0.1$, $b = 1000$, $k = 1000$ and $\lambda = 0.01$. Below, we only list the non-default hyperparameters for both GeomE2D and GeomE3D: $\lambda = 0.025$ on WN18, $\lambda = 0.05$ on FB15K-237, and $\lambda = 0.1$ on WN18RR. We implemented our model using PyTorch (Paszke et al., 2017) and ran the training processes on a single GeForce RTX 2080 GPU. To prevent over-fitting, we used the early-stop setting on validation set and set the maximum epoch to 100. The code is available at `https://github.com/soledad921/GeomE`.

Baselines We compare our models against a variety of baselines including: DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), R-GCN+ (Schlichtkrull et al., 2018), ConvE (Dettmers et al., 2018), SimplE (Kazemi and Poole, 2018), TorusE (Ebisu and Ichise, 2018), RotatE, pRotatE (Sun et al., 2019), InteractE (Vashishth et al., 2020) and QuatE^2 (Zhang et al., 2019). We choose QuatE^2 as baseline since this variant of QuatE applies N3 regularization and reciprocal approaches as our models and get the best results among all variants of QuatE. Lacroix et.al., (2018) also uses these approaches to boost the performance of ComplEx. The results reported in (Lacroix et al., 2018) are quite close to QuatE^2 regarding MRR and Hits@10. Apart from GeomE2D and GeomE3D, we test a combination model GeomE+ by utilizing the ensemble method used for DistMult (Kadlec et al., 2017) and R-GCN+ (Schlichtkrull et al., 2018) to ensemble GeomE2D and GeomE3D. A GeomE+ model consists of a GeomE2D model plus a GeomE3D model which are separately trained with the optimal hyperparameters, i.e., $\phi^{GeomE+}(h, r, t) = \phi^{GeomE2D}(h, r, t) + \phi^{GeomE3D}(h, r, t)$.

5.2 Experimental Results

Link prediction: Results on four datasets are shown in Tables 2 and 3. GeomE3D and GeomE2D, as single models, surpass other baselines on FB15K regarding all metrics. GeomE3D and GeomE2D achieve the state-of-the-art results on WN18 except Hits@10 and MR. On FB15K-237 and WN18RR where the local information is less salient, the results of GeomE3D and GeomE2D are close to QuatE^2.

	FB15K					WN18				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
DistMult*	42	0.798	-	-	0.893	655	0.797	-	-	0.946
ComplEx	-	0.692	0.599	0.759	0.840	-	0.941	0.930	0.945	0.947
ConvE	51	0.657	0.558	0.723	0.831	374	0.943	0.935	0.946	0.956
R-GCN+	-	0.696	0.601	0.760	0.842	-	0.819	0.697	0.929	**0.964**
SimplE	-	0.727	0.660	0.773	0.838	-	0.942	0.939	0.944	0.947
TorusE	-	0.733	0.674	0.771	0.832	-	0.947	0.943	0.950	0.954
RotatE	40	0.797	0.746	0.830	0.884	309	0.949	0.944	0.952	0.959
pRotatE	43	0.799	0.750	0.829	0.884	**254**	0.947	0.942	0.950	0.957
QuatE^2	-	0.833	0.800	0.859	0.900	-	0.950	0.944	0.954	0.962
GeomE2D	34	0.853	0.816	0.877	0.913	259	0.951	0.946	0.954	0.960
GeomE3D	36	0.846	0.806	0.876	0.915	325	0.951	**0.947**	0.954	0.959
GeomE+	**30**	**0.854**	**0.817**	**0.880**	**0.916**	**254**	**0.952**	**0.947**	**0.955**	0.962

Table 2: Link prediction results on FB15K and WN18. * indicates that results are taken from (Kadlec et al., 2017). Other results are taken from the original papers. Best results are written in bold.

	FB15K-237					WN18RR				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
DistMult$^\diamond$	254	0.241	0.155	0.263	0.419	5110	0.43	0.39	0.44	0.49
ComplEx$^\diamond$	339	0.247	0.158	0.275	0.428	5261	0.44	0.41	0.46	0.51
ConvE$^\diamond$	244	0.325	0.237	0.356	0.501	4187	0.43	0.40	0.44	0.52
R-GCN+	-	0.249	0.151	0.264	0.417	-	-	-	-	-
RotatE	177	0.338	0.241	0.375	0.533	3340	0.476	0.428	0.492	0.571
pRotatE	178	0.328	0.230	0.365	0.524	2923	0.462	0.417	0.479	0.552
InteracE	172	0.354	0.263	-	0.535	5202	0.463	0.430	-	0.528
QuatE^2	-	**0.366**	0.271	**0.401**	0.556	-	0.482	0.436	0.499	0.572
GeomE2D	155	0.363	0.269	0.399	0.552	3199	0.483	0.439	0.499	0.571
GeomE3D	151	0.364	0.270	0.399	0.555	3303	0.481	0.441	0.494	0.564
GeomE+	**145**	**0.366**	**0.272**	**0.401**	**0.557**	**2836**	**0.485**	**0.444**	**0.501**	**0.573**

Table 3: Link prediction results on FB15K-237 and WN18RR. $\diamond$ indicates that results are taken from (Dettmers et al., 2018). Best results are written in bold.

GeomE2D achieves the best MR, MRR and Hits@3 on WN18RR, and GeomE3D achieves the best MR on FB15K-237 as well as the best Hits@1 on WN18RR.

By combining GeomE2D and GeomE3D, GeomE+ shows more competitive performance. On FB15K, FB15K-237 and WN18RR, GeomE+ outperforms all baselines regarding all metrics. Especially on FB15K, GeomE+ improves MRR by 2.1%, Hits@1 by 1.7%, Hits@3 by 2.1% and Hits@10 by 1.6%, compared to QuatE^2. On WN18, GeomE+ achieves the state-of-the-art results regarding MR, MRR, Hits@1 and Hits@3, and achieves the second highest numbers on Hits@10.

The effect of the grade of the multivector space: Our approach can be generalized in the geometric algebras $\mathbb{G}^N$ with different grades N. In this paper, we mainly focus on GeomE models embeded in $\mathbb{G}^2$ and $\mathbb{G}^3$. We do not use multivectors with higher grade $N > 3$ in this paper because that would increase the time consumption and memory sizes of training GeomE models and the results of GeomE3D and GeomE2D on the four benchmarks are close. On the other hand, we also test the performances of GeomE1D where each multivector consists of a scalar plus a vector, and find the results drop since the 1-grade multivectors lose some algebra properties after bivectors which square is -1 are removed.

The effect of the embedding dimensionality: Figure 2 shows the link prediction results of GeomE2D models with different embedding dimensionalities $k = \{20, 50, 100, 200, 500, 1000\}$ on FB15K-237 and WN18RR regarding MRR and Hits@10. It can be seen that the performances of GeomE2D improve

	FB15K-237				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
GeomE1D	0.355	0.261	0.391	0.545	0.453	0.409	0.465	0.541

Table 4: Link prediction results of GeomE1D on FB15K-237 and WN18RR.

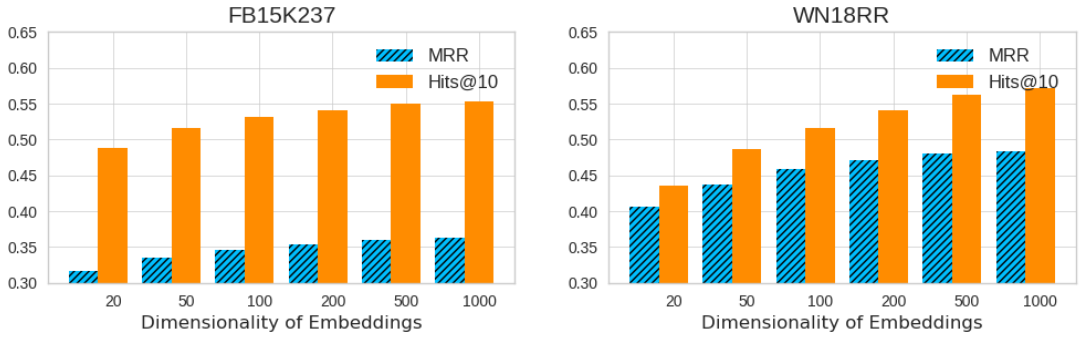

Figure 2: Results of GeomE2D with different dimensionalities on FB15K-237 and WN18RR

with the increasing of the embedding dimensionality. We follow the previous work (Zhang et al., 2019; Sun et al., 2019) to set the maximum dimensionality to 1000 in order to avoid too much memory and time consumption. It will still be interesting to explore the performances of GeomE models with higher-dimensional embeddings, e.g., Ebisu et al. (2018) use 10000-dimensional embeddings for TorusE.

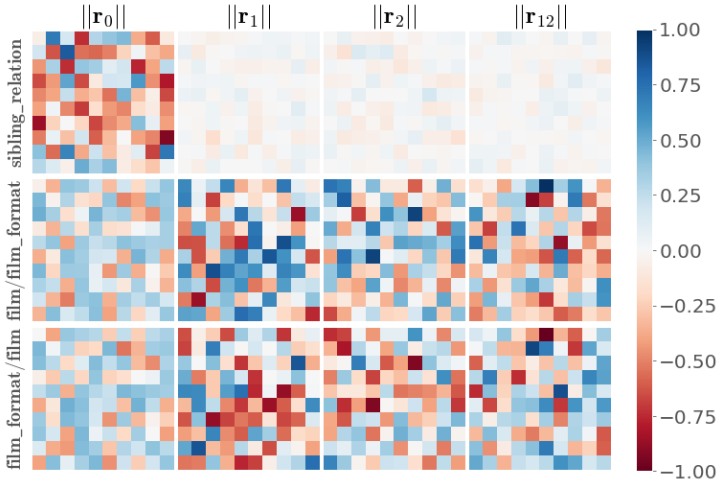

Figure 3: Visualization of the embeddings of symmetric and inverse relations. 100-dimensional embeddings are reshaped into 10×10 matrices here for a better representation.

Modeling symmetry and inversion: In FB15K, ***sibling_relationship*** is a typical symmetric relation. By constraining $\phi(h, \textbf{\textit{sibling\_relationship}}, t) \approx \phi(t, \textbf{\textit{sibling\_relationship}}, h)$ during the training process, we find that the vector and bivector parts of its embedding learned by a 100-dimensional GeomE2D are close to zero as shown in Figure 3. For a pair of inverse relations ***film/film_format*** and ***film_format/film*** in FB15K, their embeddings are matually conjugate by constraining $\phi(h, \textbf{\textit{film/film\_format}}, t) \approx \phi(t, \textbf{\textit{film\_format/film}}, h)$. These results support our arguments in Section 4.4 and empirically prove GeomE's ability of modeling symmetric and inverse relations.

6 Conclusion

We propose a new gemetric algebra-based approach for KG embedding, GeomE, which utilizes multi-vector representations to model entities and relations in a KG with the geometric product. Our approach subsumes several state-of-the-art KG embedding models, and takes advantages of the flexibility and representational power of geometric algebras to enhance its generalization capacity, enrich its expressiveness

with higher degree of freedom and enable its ability of modeling various relation patterns. Experimental results show that our approach achieves the state-of-the-art results on four well-known benchmarks.

Acknowledgements

This work is supported by the CLEOPATRA project (GA no. 812997), the German national funded BmBF project MLwin and the BOOST project.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Empirical Methods in Natural Language Processing*.

Eduardo Bayro-Corrochano. 2018. *Geometric Algebra Applications Vol. I: Computer Vision, Graphics and Neurocomputing*. Springer.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

James M Chappell, Azhar Iqbal, Lachlan J Gunn, and Derek Abbott. 2015. Functions of multivector variables. *PloS one*, 10(3).

William Kingdon Clifford. 1882. *Mathematical papers*. Macmillan and Company.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hermann Grassmann. 1844. *Die lineale Ausdehnungslehre ein neuer Zweig der Mathematik: dargestellt und durch Anwendungen auf die übrigen Zweige der Mathematik, wie auch auf die Statik, Mechanik, die Lehre vom Magnetismus und die Krystallonomie erläutert*.

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, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. 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)*, volume 1, pages 687–696.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744*.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in neural information processing systems*, pages 4284–4295.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. International Conference on Machine Learning (ICML).

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mojtaba Nayyeri, Chengjin Xu, Yadollah Yaghoobzadeh, Hamed Shariat Yazdi, and Jens Lehmann. 2019. Toward understanding the effect of loss function on the performance of knowledge graph embedding.

Dai Quoc Nguyen, Dat Quoc Nguyen, Tu Dinh Nguyen, and Dinh Phung. 2019. A convolutional neural network-based model for knowledge base completion and its application to search personalization. *Semantic Web*, 10(5):947–960.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *ICML*, volume 11, pages 809–816.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *In International Conference on Learning Representations*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML).

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119. Citeseer.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. In *In International Conference on Learning Representations*.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embedding. In *Advances in neural information processing systems*.

Appendices

Appendix A Supported Vector Space for GeomE

Table 5: Supported Vector Space.

Models	Real $\mathbb{R}$	Complex $\mathbb{C}$	Quaternion $\mathbb{H}$	Octonion $\mathbb{O}$	n-grade $\mathbb{G}^n$
TransE (Bordes et al., 2013)	✓	✗	✗	✗	✗
DistMult (Yang et al., 2014)	✓	✗	✗	✗	✗
TuckEr (Balažević et al., 2019)	✓	✗	✗	✗	✗
(p)RotatE (Sun et al., 2019)	✓	✓	✗	✗	✗
ComplEx (Trouillon et al., 2016)	✓	✓	✗	✗	✗
TransComplEx (Nayyeri et al., 2019)	✓	✓	✗	✗	✗
QuatE (Zhang et al., 2019)	✓	✓	✓	✗	✗
OctonionE (Zhang et al., 2019)	✓	✓	✓	✓	✗
GeomE	✓	✓	✓	✓	✓

Appendix B The Geometric Product of 3-grade Multivectors

The product of two 3-grade multivectors $M_a = a_0+a_1e_1+a_2e_2+a_3e_3+a_{12}e_1e_2+a_{23}e_2e_3+a_{13}e_1e_3+a_{123}e_1e_2e_3$ and $M_b = b_0 + b_1e_1 + b_2e_2 + b_3e_3 + b_{12}e_1e_2 + b_{23}e_2e_3 + b_{13}e_1e_3 + b_{123}e_1e_2e_3$ from $\mathbb{G}^3$ is represented as follows.

$$
\begin{aligned}
M_a \otimes_3 M_b =& a_0b_0 + a_1b_1 + a_2b_2 + a_3b_3 - a_{12}b_{12} - a_{23}b_{23} - a_{13}b_{13} - a_{123}b_{123} \\
&+ (a_0b_1 + a_1b_0 - a_2b_{12} + a_{12}b_2 - a_3b_{13} + a_{13}b_3 - a_{23}b_{123} - a_{123}b_{23})e_1 \\
&+ (a_0b_2 + a_2b_0 + a_1b_{12} - a_{12}b_1 - a_3b_{23} + a_{23}b_3 + a_{13}b_{123} + a_{123}b_{13})e_2 \\
&+ (a_0b_3 + a_3b_0 + a_1b_{13} - a_{13}b_1 + a_2b_{23} - a_{23}b_2 - a_{12}b_{123} - a_{123}b_{12})e_3 \\
&+ (a_0b_{12} + a_{12}b_0 + a_1b_2 - a_2b_1 - a_{13}b_{23} + a_{23}b_{13} + a_3b_{123} + a_{123}b_3)e_1e_2 \\
&+ (a_0b_{23} + a_{23}b_0 + a_1b_{123} + a_{123}b_1 + a_2b_3 - a_3b_2 - a_{12}b_{13} + a_{13}b_{12})e_2e_3 \\
&+ (a_0b_{13} + a_{13}b_0 + a_1b_3 - a_3b_1 - a_2b_{123} - a_{123}b_2 + a_{12}b_{23} - a_{23}b_{12})e_1e_3 \\
&+ (a_0b_{123} + a_{123}b_0 + a_1b_{23} + a_{23}b_1 - a_2b_{13} - a_{13}b_2 + a_3b_{12} + a_{12}b_3)e_1e_2e_3.
\end{aligned} \tag{12}
$$

Appendix C Extended Score Function of GeomE

More specifically, we define the scoring functions for GeomE2D and GeomE3D as:

$$
\begin{aligned}
&\phi^{GeomE2D}(h,r,t) = \\
&(h_0 \circ r_0 + h_1 \circ r_1 + h_2 \circ r_2 - h_{12} \circ r_{12}) \circ t_0 - (h_0 \circ r_1 + h_1 \circ r_0 - h_2 \circ r_{12} + h_{12} \circ r_2) \circ t_1 - \\
&(h_0 \circ r_2 + h_2 \circ r_0 + h_1 \circ r_{12} - h_{12} \circ r_1) \circ t_2 + (h_1 \circ r_2 - h_2 \circ r_1 + h_0 \circ r_{12} + h_{12} \circ r_0) \circ t_{12}
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
&\phi^{GeomE3D}(h,r,t) = \\
&(h_0 \circ r_0 + h_1 \circ r_1 + h_2 \circ r_2 + h_3 \circ r_3 - h_{12} \circ r_{12} - h_{23} \circ r_{23} - h_{13} \circ r_{13} - h_{123} \circ r_{123}) \circ t_0 - \\
&(h_0 \circ r_1 + h_1 \circ r_0 - h_2 \circ r_{12} + h_{12} \circ r_2 - h_3 \circ r_{13} + h_{13} \circ r_3 - h_{23} \circ r_{123} - h_{123} \circ r_{23}) \circ t_1 - \\
&(h_0 \circ r_2 + h_2 \circ r_0 + h_1 \circ r_{12} - h_{12} \circ r_1 - h_3 \circ r_{23} + h_{23} \circ r_3 + h_{13} \circ r_{123} + h_{123} \circ r_{13}) \circ t_2 - \\
&(h_0 \circ r_3 + h_3 \circ r_0 + h_1 \circ r_{13} - h_{13} \circ r_1 + h_2 \circ r_{23} - h_{23} \circ r_2 - h_{12} \circ r_{123} - h_{123} \circ r_{12}) \circ t_3 + \\
&(h_0 \circ r_{12} + h_{12} \circ r_0 + h_1 \circ r_2 - h_2 \circ r_1 - h_{13} \circ r_{23} + h_{23} \circ r_{13} + h_3 \circ r_{123} + h_{123} \circ r_3) \circ t_{12} + \\
&(h_0 \circ r_{23} + h_{23} \circ r_0 + h_1 \circ r_{123} + h_{123} \circ r_1 + h_2 \circ r_3 - h_3 \circ r_2 - h_{12} \circ r_{13} + h_{13} \circ r_{12}) \circ t_{23} + \\
&(h_0 \circ r_{13} + h_{13} \circ r_0 + h_1 \circ r_3 - h_3 \circ r_1 - h_2 \circ r_{123} - h_{123} \circ r_2 + h_{12} \circ r_{23} - h_{23} \circ r_{12}) \circ t_{13} + \\
&(h_0 \circ r_{123} + h_{123} \circ r_0 + h_1 \circ r_{23} + h_{23} \circ r_1 - h_2 \circ r_{13} - h_{13} \circ r_2 + h_3 \circ r_{12} + h_{12} \circ r_3) \circ t_{123}
\end{aligned} \tag{14}
$$

where $\circ$ denotes the Hadamard product.

Appendix D Proof of pRotatE assumption

Apart from ComplEx and QuatE, GeomE also subsumes pRotatE. We start from the formulation of the scoring function of pRotatE and show that the scoring function is a special case of Equation 8. The scoring function of pRotatE is defined as

$$\phi^{pRotatE}(h,r,t) = -\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|, \tag{15}$$

where the modulus of each element of relation vectors is $|\mathbf{r}_i| = 1, i = 1, \ldots, k$, and $|\mathbf{h}_i| = |\mathbf{t}_i| = C \in \mathbb{R}^+$. Therefore, we have

$$\begin{aligned}\phi^{pRotatE}(h,r,t) &= -\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\| = -\sqrt{\sum_{i=1}^{k} |\mathbf{h}_i\mathbf{r}_i - \mathbf{t}_i|^2} \\ &= -\sqrt{\sum_{i=1}^{k} \big(|(\mathbf{h}_i\mathbf{r}_i)|^2 + |\mathbf{t}_i|^2 - 2Re((\mathbf{h}_i\mathbf{r}_i)\bar{\mathbf{t}}_i)\big)}.\end{aligned} \tag{16}$$

Since $|\mathbf{r}_i| = 1$ and $|\mathbf{h}_i| = |\mathbf{t}_i| = C \in \mathbb{R}^+$, we have $\phi^{pRotatE}(h,r,t) = -\sqrt{2kC^2 + 2Re((\mathbf{h}_i\mathbf{r}_i)\bar{\mathbf{t}}_i)\big)}$. Considering scalars in Equation 8, i.e., $\mathbf{h}_0$, $\mathbf{r}_0$, $\mathbf{t}_0$, as real parts of complex values and bivectors $\mathbf{h}_{12}$, $\mathbf{r}_{12}$, $\mathbf{t}_{12}$ as imaginary parts of complex values, we can obtain $\phi^{GeomE2D}(h,r,t) = \frac{\phi^{pRotatE^2}(h,r,t) - 2kC^2}{2}$. Note that $2kC^2$ is a constant number as k and C, and thus does not affect the overall ranking obtained by computing and sorting the scores of triples. For a triple (h,r,t), there is a positive correlation between its GeomE score and pRotatE score since pRotatE scores are always non-positive. Therefore, GoemE2D and consequently GeomE3D *subsumes* the pRotatE model in the terms of ranking.

Appendix E Proof of Modeling various Relation Patterns in a Matrix Form

The geometric product between two 2-grade multivectors can also be represented in the form of matrix vector product as

$$M_a \otimes_2 M_b = Mat(M_a) \times vec(M_b) \times vec(e)^T = \begin{bmatrix} a_0 & a_1 & a_2 & -a_{12} \\ a_1 & a_0 & a_{12} & -a_2 \\ a_2 & -a_{12} & a_0 & a_1 \\ a_{12} & -a_2 & a_1 & a_0 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_{12} \end{bmatrix} \times \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_1e_2 \end{bmatrix}^T, \tag{17}$$

where $Mat(M_a)$ is a matrix corresponding to M_a, and $vec(M_b)$ is vector representation of M_b. Likewise, The coefficients of the geometric product of two 3-grade multivectors also can be represented as a multiplication of a 8×8 symmetric matrix and a vector.

Scoring function of GeomE can be written in the above form of matrix vector product,

$$\sum_{i}^{k} \mathbf{Sc}(M_{h_i} \otimes_n M_{r_i} \otimes_n \overline{M_{t_i}}) = \sum_{i}^{k} \langle Mat(M_{r_i}) \times vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle \tag{18}$$

where $\times$ and $\odot$ denote matrix multiplication and element-wise vector multiplication. $M_{h_i}, M_{r_i}, M_{t_i} \in \mathbb{G}^n$ are the ith multivector elements of $\mathbf{M}_h, \mathbf{M}_r, \mathbf{M}_t$, respectively. $\bar{\mathbf{1}}$ is a 2^n-dimensional vector used for changing the signs of coefficients of vector parts in an n-grade multivector ($n \leq 3$ in our case).

For a triple (h,r,t) where r is involved in an (anti-)symmetry pattern, inversion pattern or composition pattern, we import the following constraints:

$$\begin{aligned} Mat(M_{r_i}) \times vec(M_{h_i}) &= \lambda_{h_i} \cdot vec(M_{h_i}) \\ Mat(M_{r_i}) \times vec(M_{t_i}) &= \lambda_{t_i} \cdot vec(M_{t_i}). \end{aligned} \tag{19}$$

where $\cdot$ denotes scalar multiplication, λ_{h_i} and λ_{t_i} are the eigenvalues corresponding to $vec(M_{h_i})$ and $vec(M_{t_i})$.

(Anti-)Symmetry By utilizing the conjugation of embeddings of tail entities, our framework can model both two patterns. Concretely, considering equation 18 and 19, we show GeomE models symmetric pattern by $\phi(h,r,t) - \phi(t,r,h) = 0$ as follows

$$\begin{aligned}
&\mathbf{Sc}(M_{h_i} \otimes_n M_{r_i} \otimes_n \overline{M_{t_i}}) - \mathbf{Sc}(M_{t_i} \otimes_n M_{r_i} \otimes_n \overline{M_{h_i}}) = \\
&\langle Mat(M_{r_i}) \times vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle Mat(M_{r_i}) \times vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\langle \lambda_{hi} \cdot vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle \lambda_{ti} \cdot vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\lambda_{hi} \cdot \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \lambda_{ti} \cdot \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle.
\end{aligned} \tag{20}$$

Let assume that the matrix $Mat(M_{r_i})$ is householder. Therefore, it has two eigenvalues $\{-1, 1\}$. The eigenvectors corresponding to -1 are orthogonal to the eigenvalues corresponding to 1. There are two conditions

- Both of $vec(M_{t_i}), vec(M_{h_i})$ are eigenvectors (parallel) corresponding to the same eigenvalue. Therefore, we have

$$\begin{aligned}
&\lambda_{hi} \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \lambda_{ti} \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle = 0.
\end{aligned} \tag{21}$$

- Both of $vec(M_{t_i}), vec(M_{h_i})$ are eignevectors (orthogonal) corresponding to different eigenvalues. Therefore, we have

$$\begin{aligned}
&\lambda_{hi} \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \lambda_{ti} \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle + \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle = 2 \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle.
\end{aligned} \tag{22}$$

  The abovementioned equation equals to zero if for either $vec(M_{h_i})$ or $vec(M_{t_i})$, the elements (coefficents of vector parts in this case) corresponding to the negative sign (-1) of $\bar{\mathbf{1}}$ will be zero.

The abovementioned conditions hold for each multivector element M_i of the k-dimensional embeddings $\mathbf{M}$, i.e. $i = 1, \ldots, k$. Therefore, there are 2^k possible options (capacity of model) to have $\phi(h,r,t) - \phi(t,r,h) = 0$ (modeling symmetric pattern).

Inversion Given two relations r_1, r_2 which form inverse pattern i.e. $r_1 = r_2^{-1}$ (e.g. r_1 =*SonOf*, r_2 =*FatherOf*), we show that GeomE models inverse pattern by $\phi(h,r_1,t) - \phi(t,r_2,h) = 0$. This is proved as follows

$$\begin{aligned}
&\mathbf{Sc}(M_{h_i} \otimes_n M_{r_{1i}} \otimes_n \overline{M_{t_i}}) - \mathbf{Sc}(M_{t_i} \otimes_n M_{r_{2i}} \otimes_n \overline{M_{h_i}}) = \\
&\langle Mat(M_{r_{1i}}) \times vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle Mat(M_{r_{2i}}) \times vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\langle \lambda_{1hi} \cdot vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle \lambda_{2ti} \cdot vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\lambda_{1hi} \cdot \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \lambda_{2ti} \cdot \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle.
\end{aligned} \tag{23}$$

Let assume that the matrices of $Mat(M_{r_{1i}})$ and $Mat(M_{r_{2i}})$ have same eigenvalues $\lambda_{1i} = \lambda_{2i}$. Therefore, we have

$$\begin{aligned}
&\lambda_{1hi} \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \lambda_{2ti} \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle = \\
&\lambda_{1hi} (\langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle vec(M_{t_i}), vec(M_{h_i}) \odot \bar{\mathbf{1}} \rangle) = \\
&\lambda_{1hi} (\langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle - \langle vec(M_{h_i}), vec(M_{t_i}) \odot \bar{\mathbf{1}} \rangle) = 0.
\end{aligned} \tag{24}$$

Since for n-grade multivector, there are 2^n variables in the vector, the corrsponding matrices $Mat(M_{r_{1i}}), Mat(M_{r_{2i}})$ are $2^n \times 2^n$ dimensional. Since a $2^n \times 2^n$ matrix has at most 2^n distinct

eigenvalues/eigenvectors, GeomE with k dimension for embedding can at most represent 2^{n^k} distinct entity embedding vectors (model capacity) for encoding inverse pattern.

Composition By enforcing the coefficents of vector parts of $\mathbf{M}_h$ and $\mathbf{M}_t$ to be zero, we obtain

$$\begin{aligned} &\textbf{Sc}(M_{h_i} \otimes_n M_{r_i} \otimes_n \overline{M_{t_i}}) = \langle Mat(M_{r_i}) \times vec(M_{h_i}), vec(M_{t_i}) \rangle \\ &\text{subject to:} \quad h_1^i, \cdots, h_n^i = 0 \\ &\qquad\qquad\quad t_1^i, \cdots, t_n^i = 0 \end{aligned} \tag{25}$$

where $h_1^i, \cdots, h_n^i$ and $t_1^i, \cdots, t_n^i$ are the vector parts of M_{h_i} and M_{t_i}, $n \leq 3$ in our case.

Since $vec(M_{h_i})$ and $vec(M_{t_i})$ are eigenvectors of $Mat(M_{r_i})$ (defined in Equation 19), the maximization of $\textbf{Sc}(M_{h_i} \otimes_n M_{r_i} \otimes_n \overline{M_{t_i}})$ is equivalent to minimizing $\angle(Mat(M_{r_i}) \times vec(M_{h_i}), vec(M_{t_i}))$ under the following constraints,

$$\begin{aligned} &\text{arg max } \textbf{Sc}(M_{h_i} \otimes_n M_{r_i} \otimes_n \overline{M_{t_i}}) = \text{arg max } \langle Mat(M_{r_i}) \times vec(M_{h_i}), vec(M_{t_i}) \rangle = \\ &\text{arg min } \angle(Mat(M_{r_i}) \times vec(M_{h_i}), vec(M_{t_i})) \\ &\text{subject to:} \quad Mat(M_{r_i}) \times vec(M_{h_i}) = \lambda_{h_i} \cdot vec(M_{h_i}) \\ &\qquad\qquad\quad Mat(M_{r_i}) \times vec(M_{t_i}) = \lambda_{t_i} \cdot vec(M_{t_i}). \\ &\qquad\qquad\quad h_1^i, \cdots, h_n^i = 0 \\ &\qquad\qquad\quad t_1^i, \cdots, t_n^i = 0 \end{aligned} \tag{26}$$

Note that eigenvectors are scale-free. It means given $A \times x = \lambda \cdot x$ where A is a matrix, x and λ are its eigenvector and the corresponding eigenvalue, the multiplication of λ by any real number c is also an eigenvector, i.e., $A \times (c \cdot x) = \lambda \cdot c \cdot x$. Therefore, maximization of $\langle Mat(M_{r_i}) \times vec(M_{h_i}), vec(M_{t_i}) \rangle$, is equivalent to minimizing the angle between two vectors under the scale-free condition. Based on the above-mentioned assumption, and in order to model composition pattern $r_1(h, o) \wedge r_2(o, t) \Rightarrow r_3(h, t)$, we have

$$\begin{aligned} Mat(M_{r1i}) \times vec(M_{hi}) &= \alpha_1 \cdot vec(M_{oi}), \\ Mat(M_{r2i}) \times vec(M_{oi}) &= \alpha_2 \cdot vec(M_{ti}), \\ Mat(M_{r3i}) \times vec(M_{hi}) &= \alpha_3 \cdot vec(M_{ti}), \end{aligned} \tag{27}$$

where $\alpha_1, \alpha_2, \alpha_3$ are real numbers. Furthermore, we obtain the following connection between embeddings of r_1, r_2 and r_3.

$$Mat(M_{r_3 i}) = \alpha \cdot Mat(M_{r_1 i}) \times Mat(M_{r_2 i}) \tag{28}$$

where $\alpha = \alpha_3/(\alpha_1 \cdot \alpha_2)$ can be any real number. Thus, $\forall h, o, t \;\; r_1(h, o) \wedge r_2(o, t) \Rightarrow r_3(h, t)$ holds true under the constraints in Equation 19 when Equation 28 is valid.

Appendix F Dataset

Datasets	#Entities	#Relations	#Training	#Validation	#Training
FB15K	14951	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 6: Number of entities, relations, and observed triples in each split for four benchmarks.