

DensE: An Enhanced Non-Abelian Group Representation for Knowledge Graph Embedding

Haonan Lu*

AARC

Huawei Technologies

luhaonan@huawei.com

Hailin Hu

AARC

Huawei Technologies

huhailin2@huawei.com

Abstract

Capturing the composition patterns of relations is a vital task in knowledge graph completion. It also serves as a fundamental step towards multi-hop reasoning over learned knowledge. Previously, rotation-based translational methods, e.g., RotatE, have been developed to model composite relations using the product of a series of complex-valued diagonal matrices. However, RotatE makes several oversimplified assumptions on the composition patterns, forcing the relations to be commutative, independent from entities and fixed in scale. To tackle this problem, we have developed a novel knowledge graph embedding method, named DensE, to provide sufficient modeling capacity for complex composition patterns. In particular, our method decomposes each relation into an $SO(3)$ group-based rotation operator and a scaling operator in the three dimensional (3-D) Euclidean space. The advantages of our method are twofold: (1) For composite relations, the corresponding diagonal relation matrices can be non-commutative and related with entity embeddings; (2) It extends the concept of RotatE to a more expressive setting with lower model complexity and preserves the direct geometrical interpretations, which reveals how relations with distinct patterns (i.e., symmetry/anti-symmetry, inversion and composition) are modeled. Experimental results on multiple benchmark knowledge graphs show that DensE outperforms the current state-of-the-art models for missing link prediction, especially on composite relations.

1 Introduction

Knowledge graphs (KGs) are a key component of a wide range of down-stream applications, such as machine reasoning, information retrieval and knowledge-guided natural language processing [4, 19, 16, 6]. Especially, learning how to hop over a variety of concepts or instances stored in knowledge graph represents a value path towards artificial general intelligence.

Mathematically, knowledge graphs are defined as a collection of triplets. Each triplet, denoted by (h, r, t) , indicates a relation r pointing from the head entity h to tail entity t . Currently, numerous research efforts have been devoted to developing knowledge graph embedding (KGE) methods. These methods aim to learn a set of low-dimensional representations of entities and relations [4, 10], which is usually coupled with a score function to enable the knowledge graph completion process, i.e., predicting missing links between entities, for real-world KGs [11, 5, 1, 13, 18]. Sometimes, neural networks can be inserted into the process [2, 12, 9], though this requires additional computation costs.

Recently, modeling relation as rotational operators has emerged as a promising approach for KGE. For example, RotatE [13] first introduced relational rotation in a translational KGE method, which views relations as rotation operations of entities in a 2-D complex space. While this method has

*Corresponding Author. These two authors contributed equally.

shown prominent modeling power for various relation patterns (e.g., symmetry, anti-symmetry and inversion), the modeling of composition patterns is actually over-simplified. In particular, the non-commutative composite relations are barely attended to [3]. In addition, the rotation axis is forced to be orthogonal to the entity embeddings, ruling out the interaction with relation and entity embeddings. A more recent model, called QuatE [18], attempts to increase the expressiveness by conducting relational rotation as quaternion multiplication in hypercomplex space. However, its dependency on unit quaternion-based rotation still limits the modeling capacity for general composition patterns and the rotation in quaternion space fails to provide an understandable geometric interpretation to be presented.

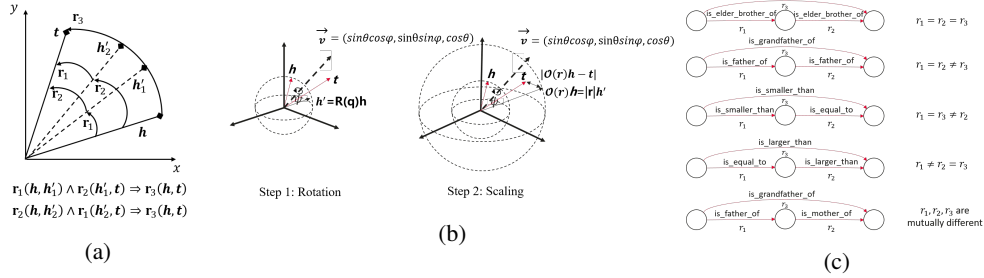


Figure 1: (a) In RotatE model, composite relations are assumed to be commutative. Changing the order of relational rotation of r_1 and r_2 gives the same composition r_3 . Also, the rotation axis (perpendicular to the paper) is orthogonal to entity embeddings. (b) DensE decomposes a relation into a rotation operator and a scaling operator on the head entity h in 3-D Euclidean space. (c) Examples of composition patterns.

In this work, we focus on the modeling of composite relations, accounting for their possible non-commutative nature and interactions with entity embeddings. More specifically, we propose DensE (**D**istance-based **E**MBEDDING with **N**on-abelian **R**otation and **S**caling in 3-D **E**UCLIDEAN **S**PACE), which decomposes the relation into an $SO(3)$ group-based rotation operator and a scaling operator and in the 3-D Euclidean space. Intuitively, non-Abelian group (here we use $SO(3)$ rotation group) is applied to introduce non-commutative nature to our model, and the scaling operation offers another important dimension to accommodate each triplet in the Euclidean space, which is barely explored in previous research. Our main contributions are summarized as the following: (1) For the first time, relational rotation and scaling are modeled within a unified framework, which achieves a much higher expressiveness with lower model complexity; (2) Our method provides an up-to-date most comprehensive while straightforward geometric interpretation of each relation type in 3-D Euclidean space; (3) Extensive experiments show that DensE outperforms the current state-of-the-art models in link prediction, offering a useful tool for knowledge graph completion.

2 Related Work

From the perspective of modeling composite relations (i.e., a relation path composed of a series of relations), our work is related to several previous methods, i.e., TransE [1], RotatE [13] and QuatE [18]. TransE and RotatE are both translational models that use Euclidean distance as score function. While they are motivated by clear geometric interpretations, their modeling capacity for composite relations is insufficient because both of them assume a commutative pattern on relation path and do not consider entity information in inferring composition patterns. Specifically, TransE models each relation as a pure translational transformation, so it assumes a fixed addition composition pattern between relations, i.e., $r_3 = r_1 + r_2$, which is commutative and irrelevant to entity embeddings. RotatE made significant progress by modeling relations as rotational operator (rotation matrix) in 2-D Euclidean space. When modeling a relation path composed of multiple relations, RotatE uses Hadamard product to combine the rotation matrices of the relations on the path, i.e., $r_3 = r_1 \circ r_2$. In this model, all relations in the composite relation have the same rotation axis (Figure 1(a)). Thus, the compositions in RotatE are also mandatorily commutative. Also, interactions between relation and entity embeddings are precluded as the rotation axis is always perpendicular to entity embeddings.

To offer a more expressive model for composite relations, QuatE is designed to represent each relation type as a rotational operator in quaternion space. While this method does not assume any fixed composition pattern, QuatE still has two major limitations. Firstly, as discovered in the original paper,

QuatE requires normalization of relation to unit quaternion, indicating it is incapable of integrating scale information (also the case in RotatE). Secondly, as both entities and relations are embedded in quaternion hyperplanes, QuatE cannot provide a straightforward geometric interpretation in the space, which hinders the understanding of the learned embeddings. Concurrent with our work, [17] proposes a group theoretic analysis for KGE methods. Their method, named NagE, represents a preliminary attempt in applying non-Abelian group in modeling relational rotations. However, their empirical results show limited performance advance over previous methods (e.g., RotatE). Also, their method does not provide sufficient geometric insights in the modeling process.

In contrast to the previous works, our model leverages both rotation and scaling operations for relation modeling. The key idea is that we can transform any non-zero vector in the Euclidean space to another arbitrary vector through decoupled rotation and scaling transformations. In addition, our model provides a clear geometric picture to demonstrate the transformation of entity representation in various relation composition patterns.

3 Problem Formulation

We denote a directed knowledge graph as $G(\mathcal{E}, \mathcal{R}, \mathcal{F})$, where \mathcal{E} , \mathcal{R} and \mathcal{F} are sets of entities, relations and facts, respectively. A fact stored in a KG can be expressed as a triplet $(h, r, t) \in \mathcal{F}$, where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. Herein, we focus on the knowledge completion task, which aims to predict missing links based on the observed facts. To fulfill this goal, a score function is used to measure the plausibility of proposed fact candidates, and the goal of model optimization is to give higher scores to true triplets (h, r, t) than the false triplets (h, r, \bar{t}) or (\bar{h}, r, t) , where \bar{t} and \bar{h} are randomly sampled tail and head entities, respectively. Mathematically, the entity and relation embeddings are usually represented by tensors, and the score function can thus be written into the form of $f_r(\mathbf{h}, \mathbf{t})$.

In principle, KGE models should be designed to accommodate various relation patterns existing in real world KGs, such as **symmetry** ($r(x, y) \Rightarrow r(y, x)$), **anti-symmetry** ($r(x, y) \Rightarrow \neg r(y, x)$) and **inversion** ($r_1(x, y) \Rightarrow r_2(y, x)$). Formal definitions and examples of these patterns can be found in the **Appendix**. In contrast to the above atomic relation patterns (inferable within one hop), the complex composition patterns pose a particular challenge to modeling, as discussed below.

Definition 1. Relation r_3 is **composed** of relation r_1 and relation r_2 if $\forall x, y, z$

$$r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z) \quad (1)$$

Here r_3 is also referred to as a **composite relation** and possesses certain **composition pattern**. In particular, our model design takes the following properties into account.

Property 1. The two relations in the composition are not always **commutative**. For example, given $r_1 = \text{is_father_of}$, $r_2 = \text{is_mother_of}$, based on the **Definition 1**, we will get $r_3 = \text{is_grandfather_of}$. However, when we change the order, i.e., $r_1 = \text{is_mother_of}$, $r_2 = \text{is_father_of}$, we will get $r_3 = \text{is_grandmother_of}$.

Property 2. The composition patterns are not always inferable by the relations alone. For example, given that y is x 's younger sister and z is y 's elder brother, we can not answer whether z is elder or younger than x from the given information. Actually, to answer this question, we need to know more about $x/y/z$ from their own attributes and their other relationships.

Property 3. In a composition, the relations involved are not necessarily different. Given the two-hop example above, besides the situation that r_1 , r_2 and r_3 are mutually different, there are also four different cases that satisfy the definition of composition, i.e., $r_1 = r_2 = r_3$, $r_1 = r_2 \neq r_3$, $r_1 = r_3 \neq r_2$ and $r_2 = r_3 \neq r_1$ (Figure 1(c)).

4 Method

4.1 Modeling relational rotation using SO(3) rotation group

One of the ways to model a rotation operation in the 3-D space is called axis-angle representation, which parameterizes a rotation by two quantities: 1) A unit vector \vec{v} indicating the direction of the axis of rotation, i.e., $\vec{v} = (v_x, v_y, v_z) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$, where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$; and 2) An angle ψ describing the magnitude of the rotation about the rotation axis, where

$\psi \in [0, 2\pi)$. Given an entity vector \vec{w} in the 3-D space with the coordinate (x, y, z) , its rotation about axis \vec{v} with an angle of ψ can be modeled using the SO(3) group theory (Figure 1(b), Step 1). More specifically, we can use a unit quaternion to encode the rotation using three degrees of freedom (i.e., θ , ϕ and ψ). Actually, it can be viewed as a group structure on a 3-sphere (i.e., S^3) which gives the group Spin(3). Note that this group structure is isomorphic to SU(2) group and also to the universal cover of SO(3) group. Formally, the unit quaternion \mathbf{q} to model a rotation through an angle of ψ around the aforementioned axis \vec{v} can be derived using an extension of Euler's formula:

$$\mathbf{q} = e^{\frac{\psi}{2}(v_x\mathbf{i}+v_y\mathbf{j}+v_z\mathbf{k})} = \cos \frac{\psi}{2} + \sin \frac{\psi}{2} * (v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}), \quad (2)$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are imaginary units of the quaternion representation, which satisfies the condition $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. Unlike real/complex numbers, the multiplication of quaternions (Hamilton product) is sensitive to the orders as we have: $\mathbf{ij} = \mathbf{k}, \mathbf{ji} = -\mathbf{k}, \mathbf{jk} = \mathbf{i}, \mathbf{kj} = -\mathbf{i}, \mathbf{ki} = \mathbf{j}, \mathbf{ik} = -\mathbf{j}$. For $\mathbf{Q}_1 = a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}$ and $\mathbf{Q}_2 = a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}$, their Hamilton product is:

$$\begin{aligned} \mathbf{Q}_1 \otimes \mathbf{Q}_2 = & a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2 + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)\mathbf{i} \\ & + (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)\mathbf{j} + (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)\mathbf{k} \end{aligned} \quad (3)$$

A 3-D Euclidean vector \vec{w} with the coordinate (x, y, z) can be expressed as a pure quaternion (meaning the real part of quaternion is zero), i.e., $\mathbf{W} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$, giving the following theorem:

Theorem 1 Given a 3-D Euclidean vector \vec{w} and its counterpart in the quaternion space \mathbf{W} , the desired rotation axis \vec{v} , the magnitude of the rotation ψ , the destination coordinate of the vector after the rotation, i.e., $\mathbf{W}' = x'\mathbf{i} + y'\mathbf{j} + z'\mathbf{k}$, can be calculated by the Hamilton product of quaternions:

$$\mathbf{W}' = \mathbf{q}\mathbf{W}\mathbf{q}^{-1} \quad (4)$$

where \mathbf{q}^{-1} is the inverse of \mathbf{q} , i.e., $\mathbf{q}^{-1} = e^{-\frac{\psi}{2}(v_x\mathbf{i}+v_y\mathbf{j}+v_z\mathbf{k})} = \cos \frac{\psi}{2} - \sin \frac{\psi}{2} * (v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k})$.

The form of Eq.4 and a factor of $\frac{1}{2}$ for the angle ψ in Eq.2 indicate that there is a 2 : 1 homomorphism from quaternions of unit norm to SO(3). Considering each 3-D Euclidean vector can also be expressed as a pure quaternion, we can now represent the rotation using a matrix $\mathbf{R}(\mathbf{q})$ by expanding Eq.4 and letting $C = \cos \psi$ and $S = \sin \psi$:

$$\vec{w}' = \mathbf{R}(\mathbf{q})\vec{w} = \begin{bmatrix} C + v_x^2(1-C) & v_xv_y(1-C) + v_zS & v_xv_z(1-C) - v_yS \\ v_xv_y(1-C) - v_zS & C + v_y^2(1-C) & v_yv_z(1-C) + v_xS \\ v_xv_z(1-C) + v_yS & v_yv_z(1-C) - v_xS & C + v_z^2(1-C) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (5)$$

In our framework, two rotations can be combined into one equivalent rotation operation (this is also consistent with the closure property of group theory). In other words, we can define $\mathbf{q} = \mathbf{q}_2\mathbf{q}_1$, where \mathbf{q} corresponds to the rotation \mathbf{q}_1 followed by the rotation \mathbf{q}_2 . Therefore, a series of rotations can be **composed** together and then applied as a single rotation. Note that quaternion multiplication is not commutative unless \mathbf{q}_1 and \mathbf{q}_2 share the same rotation axes (i.e., $\vec{v}_1 = \vec{v}_2$), which can be seen from Eq.3. This makes it possible to model both commutative and non-commutative relation patterns. The geometrical interpretation of non-commutative modeling can be found in **Appendix**.

4.2 Integrating scaling operations

Following Eq.2 and letting \mathbf{q} to be the unit quaternion, an arbitrary quaternion with non-unit norm can be written as: $\mathbf{Q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = |Q|\mathbf{q}$, with the norm given by $|Q| = \sqrt{a^2 + b^2 + c^2 + d^2}$ where $a = |Q| \cos \frac{\psi}{2}$, $b = |Q| \sin \frac{\psi}{2} \sin \theta \cos \phi$, $c = |Q| \sin \frac{\psi}{2} \sin \theta \sin \phi$ and $d = |Q| \sin \frac{\psi}{2} \cos \theta$. By multiplying a scalar $|Q|$ in the Eq.5, we can further introduce length as another degree of freedom to better match the ground-truth tail embedding vector (Figure 1(b), Step 2). Formally, we have:

$$\vec{w}' = |Q|\mathbf{R}(\mathbf{q})\vec{w} = \mathcal{O}(\mathbf{Q})\vec{w}, \text{ where } \mathbf{Q} \in \mathbb{H}, \vec{w}, \vec{w}' \in \mathbb{R}^3, \quad (6)$$

where $\mathcal{O}(\mathbf{Q}) = |Q|\mathbf{R}(\mathbf{q})$ is the combined operator of rotation and scaling transformations, \mathbb{H} denotes quaternion algebra, and \mathbb{R}^3 represents 3-D Euclidean algebra. Here we call $|Q|$ the *scaling factor*.

Therefore, we now have a uniform framework with interpretable geometric meaning, i.e., $(|Q|, \theta, \phi, \psi)$ to describe the transformation corresponding to a specific relation type. We can also define the reverse operation $\mathcal{O}(\mathbf{Q}^{-1}) = |Q|^{-1}\mathbf{R}(\mathbf{q}^{-1})$, which describes the reverse process: rotate a vector about the axis \vec{v} with angle $-\psi$ (from another direction), and then scale the vector with a factor of $|Q|^{-1}$. Combining the Eq.5 and Eq.6, we can always find an operator $\mathcal{O}(\mathbf{Q}_3) = \mathcal{O}(\mathbf{Q}_2)\mathcal{O}(\mathbf{Q}_1)$, which corresponds to the application of $\mathcal{O}(\mathbf{Q}_1)$ followed by the application of $\mathcal{O}(\mathbf{Q}_2)$, where we have $|Q_3| = |Q_1| * |Q_2|$ and $\mathbf{R}(\mathbf{q}_3) = \mathbf{R}(\mathbf{q}_2)\mathbf{R}(\mathbf{q}_1)$.

4.3 Score function and optimization

A score function aims to correctly measure the plausibility of a triple of interest. Formally, as a distance-based model, our scoring function is defined as $f_r(\mathbf{h}, \mathbf{t}) = -\frac{1}{2}(|\mathcal{O}(\mathbf{r})\mathbf{h} - \mathbf{t}| + |\mathcal{O}(\mathbf{r}^{-1})\mathbf{t} - \mathbf{h}|)$. Here, $|\cdot|$ denotes the Euclidean distance and $\mathcal{O}(\cdot)$ stands for the transformation conducted on each element of the entity embeddings, i.e., $\mathbf{t}_i = \mathcal{O}(\mathbf{r}_i)\mathbf{h}_i$, and $\mathbf{h}_i = \mathcal{O}(\mathbf{r}_i^{-1})\mathbf{t}_i$, where $\mathbf{r}_i \in \mathbb{H}$, $\mathbf{h}_i, \mathbf{t}_i \in \mathbb{R}^3$, and i indicates i -th embedding unit of \mathbf{h} . The arrow of \mathbf{h}_i and \mathbf{t}_j are omitted for clarity. To properly train the model parameters, here we use a loss function similar to the self-adversarial negative sampling loss proposed in [13]:

$$L = -\log \sigma(\gamma + f_r(\mathbf{h}, \mathbf{t})) - \sum_{j=1}^n p(\bar{h}_j, r, \bar{t}_j) \log \sigma(-(\gamma + f_r(\bar{\mathbf{h}}^{(j)}, \bar{\mathbf{t}}^{(j)}))), \quad (7)$$

where γ is a fixed margin, n is the number of negative sampling size, $(\bar{h}_j, r, \bar{t}_j)$ is the j -th negative triplet of the fact (h, r, t) , and σ is the sigmoid function. $\bar{\mathbf{h}}^{(j)}$ and $\bar{\mathbf{t}}^{(j)}$ are the embeddings corresponding to the negative triplet $(\bar{h}_j, r, \bar{t}_j)$. $p(\bar{h}_j, r, \bar{t}_j)$ is the weight of the negative sample, which gives the higher scored negative samples with larger weight during training. The details about self-adversarial negative sampling technique can be found in [13].

4.4 Capability of DensE in modeling symmetry, antisymmetry, inversion and composition

Enabled by the geometric interpretation provided by each degree of freedom of our model, here we discuss how DensE models the aforementioned relation patterns. Clearly, a relation r is **symmetric** in DensE if and only if each dimension of its embedding \mathbf{r}_i satisfies $|\mathbf{r}_i| = 1$ and the rotation angle satisfies $\psi_{r_i} = 0$ or π . For **anti-symmetry** relation pattern, the embedding \mathbf{r}_i satisfies $|\mathbf{r}_i| = 1$, but the rotation angle ψ_{r_i} should be neither 0 nor π . Also, two relations r_1 and r_2 are in **inverse** pattern, if and only if they satisfy: $|\mathbf{r}_{1i}| * |\mathbf{r}_{2i}| = 1$, $\theta_{r_{1i}} = \theta_{r_{2i}}$, $\phi_{r_{1i}} = \phi_{r_{2i}}$ and $\psi_{r_{1i}} = -\psi_{r_{2i}}$, meaning the embeddings of these two relations share the same rotation axes, but rotate in two opposite directions.

As discussed in **Section 4.1**, the **commutative** and **non-commutative composition** patterns can be naturally modeled by the guarantee of the property of group theory, which covers **Property 1** of composite relations. Also, following the intuition of **Property 2**, our model does not enforce a uniform mode of each element in relation representations. Instead, it learns to model the interaction between relations and entities as well as the ambiguity in composition pattern inference, leading to a disperse distribution in the relation embedding space. Last but not least, our model can smoothly deal with constraints posed by relation types in inferring composition patterns, as stated in **Property 3**. For instance, when modeling the pattern $r_1(x, y) \wedge r_2(y, z) \Rightarrow r_2(x, z)$, the representation from RotatE tends to degenerate to a trivial case where the rotation angle of r_1 and r_2 both set to be 0 or 2π or $r_1 = 2\pi, r_2 = \pi$. In DensE, since the entity embeddings are not required to be perpendicular to the rotation axis, it can also place the embedding of entity x to be collinear with the rotation axis of r_1 . In this way, the rotation axis of r_1 and r_2 are not required to be the same, making the model to be more expressive. In another example, as for the pattern $r_1(x, y) \wedge r_1(y, z) \Rightarrow r_2(x, z)$, besides capturing the relationship of the two rotation angles (i.e., $\psi_{r_{2i}} = 2\psi_{r_{1i}}$), the scaling transformation offers an additional degree of freedom, where our model tends to give $|\mathbf{r}_{2i}| = |\mathbf{r}_{1i}|^2$. Again, we point out that these ‘‘rules’’ are not constant solutions, as the information entities will further guide the model to deviate from the statistical mode for a better accommodation of each triplet. Other composition patterns presented in the **Property 3** can be analyzed in the similar way (see **Appendix**).

Table 1: Performance comparison on benchmark datasets. Best results are labeled in bold and the second best are underlined. Performance of TransE is from [11] and [10]; Performance of ComplEx is from [2]; Performance of RotatE, QuatE and NagE are taken from the corresponding original papers. QuatE and QuatE⁺ denote the plain QuatE model and QuatE model with N3 regularization and reciprocal learning, respectively. Model performance in other evaluation metrics can be found in the **Appendix**.

Model	WN18		WN18RR		FB15K-237		YAGO3-10	
	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
TransE	0.495	0.113	0.226	-	0.294	-	-	-
ComplEx	0.941	0.936	0.44	0.41	0.247	0.158	0.36	0.26
RotatE	0.949	0.944	0.476	0.428	0.338	0.241	<u>0.495</u>	<u>0.402</u>
NagE	0.950	<u>0.944</u>	0.477	0.432	0.340	0.244	-	-
QuatE	<u>0.949</u>	0.941	0.481	<u>0.436</u>	0.311	0.221	-	-
QuatE ⁺	0.950	<u>0.944</u>	<u>0.482</u>	<u>0.436</u>	0.366	0.271	-	-
DensE	0.950	0.945	0.491	0.443	<u>0.349</u>	<u>0.256</u>	0.541	0.465

5 Experiments

Datasets and evaluation metrics The experiments are conducted on four commonly used benchmark datasets, including WN18, WN18RR, Fb15k-237 and YAGO3-10. WN18 comes from WordNet [8], which consists of many lexical relations of English words. As WN18 has a significant part of inverse relations, it mainly reflects inversion pattern of relations. Later, WN18RR [2] is proposed where inverse relations are removed. Similarly, the FB15k-237 dataset [14] is extracted from the original Freebase dataset FB15K [1] by removing inverse relations. In addition, we also use the YAGO3-10 [7] dataset, which consists of a large collection of triplets from multilingual Wikipedia. The latter three datasets aim to assess the model performance on composition patterns. Here, we report mean reciprocal rank (MRR) and Hits at 1 (H@1) for evaluation (the higher, the better). Statistics of these datasets and experiments in other evaluation metrics (mean rank, H@3 and H@10) are provided in the **Appendix**.

Implementation details We use the Adam optimizer and tune the hyperparameters on the validation dataset. During training, we adopt a similar reciprocal learning approach as used in [5, 18]. Early stopping is applied based on the performance on validation dataset every 1000 epochs. The ranges for hyperparameter grid search and the best hyperparameter settings are listed in **Appendix**. All parameters are randomly initialized from the interval $[-\frac{1}{\sqrt{2k}}, \frac{1}{\sqrt{2k}}]$, where k is the embedding size.

Baselines We mainly compare DensE with top-performing baseline models for KG link prediction, including TransE [1], ComplEx [15], RotatE [13], NagE [17] and QuatE [18]. These baselines have included both translational model (TransE, NagE and RotatE) and semantic matching model (ComplEx and QuatE). Note that in the original QuatE paper, the authors reported the results in three settings. One of them involves a “type constraint” operation which altered standard evaluation protocol. To ensure fairness, we do not include this setting for comparison.

6 Results and Analysis

Prediction performance We report the link prediction results on the four benchmark datasets in Table 1. Expectedly, on WN18 dataset, due to the widely existing atomic relations, all the models show comparably high performance. In contrast, when the inverse relations are removed, we observe a significant margin between most of the compared methods, except TransE, which cannot model symmetric relations. In particular, we show that DensE outperforms all the other models on all metrics. To confirm the source of performance gains, we conduct a further analysis that compares the MRR performance of DensE to RotatE on each relation type, as listed in Table 2. Intriguingly, we notice a large performance increase on composite relations, as exemplified by *hypernym*, the most abundant composite relation in test data. We show that DensE improves MRR on this relation by as much as 3.3%. These results indicate a particular advantage of DensE in modeling composition relation patterns. On FB15k-237, we show that the performance of DensE is superior to most of

the baseline models, including RotatE and plain version of QuatE. On this specific dataset, we also notice QuatE using N3 regularization and reciprocal learning (QuatE⁺) performs especially well. However, this improvement increases the parameter size of QuatE by a factor of ten (5.8M to 58.2M), which greatly hinders its application on large scale dataset. On YAGO3-10, DensE also shows a large margin over RotatE and ComplEx, which further demonstrates the superiority of DensE on various types of datasets. Results by other evaluation metrics can be found in the **Appendix**.

Model complexity As listed in Table 3, we show that compared with high-performance models such as RotatE and QuatE⁺, DensE has much lower model complexity in terms of parameter number and training epochs. We reason that this is mainly achieved by introducing a decoupled scaling operation, thus lowering the embedding dimension required in rotation-only modeling.

Table 2: MRR comparison on each relation type of WN18-RR. Performance increases are in parentheses.

Relation type	Relation Name	% in test data	RotatE	DensE
Atomic	derivationally_related_form	34%	0.947	0.955 (+0.008)
	also_see	1.8%	0.585	0.647 (+0.062)
	verb_group	1.3%	0.943	0.955 (+0.012)
	similar_to	0.2%	1	1 (+0)
Composite	hypernym	39.5%	0.148	0.181 (+0.033)
	instance_hyponym	4%	0.318	0.349 (+0.031)
	member_meronym	8.1%	0.232	0.249 (+0.017)
	synset_domain_topic_of	3.8%	0.341	0.412 (+0.071)
	has_part	5.5%	0.184	0.205 (+0.021)
	member_of_domain_usage	0.8%	0.318	0.326 (+0.008)
	member_of_domain_region	1%	0.2	0.407 (+0.207)

Table 3: Model complexity (#parameters/#training epochs). Values obtained from the best configurations reported in the original paper. Training epochs of RotatE are obtained by re-implementation.

Model	RotatE	QuatE ⁺	DensE
Space	\mathbb{C}^k	\mathbb{H}^k	$\mathbf{h}, \mathbf{t} \in \mathbb{R}^{3 \times k}, \mathbf{r} \in \mathbb{H}^k$
WN18	41.0M/48K	163.8M/-	24.6M/18K
WN18RR	41.0M/45K	163.8M/-	24.6M/15K
FB15k-237	29.3M/100K	58.2M/-	22.0M/27K
YAGO3-10	123.2M/90K	-	73.9M/33K

Geometric interpretation Here we show several examples to illustrate the geometric insight given by DensE, which basically reflects the geometric intuition discussed in **Section 4.4**. In Figure 2(a), we first show the distribution of element-wise addition of embeddings from two **inverse** relations (*has_part* and *part_of*) of ψ , one representative degree of freedom in modeling relational rotation (Other degrees of freedom can be found in the **Appendix**). In this way, we can visualize how the two embeddings agree with each other. As the two relations are fully inferable by each other, we do observe a clear conjugation as expected in **Section 4.4**. This is also reflected in the representation of scaling from these two relations, where the element-wise products tend to be one (Figure 2(b)). Interestingly, we do observe two different embeddings are learned for each relation (Figure 2(c)-(d)), demonstrating the scaling operation do capture relation-specific features. For **composition patterns**, we slightly change the experiment protocol, with each histogram showing the element-wise difference between the embeddings of a composite relation and the embeddings calculated by multiplying each relation in the relation path. In a case from WN18RR, we demonstrate how DensE models a composition pattern for $r_1(h, h') \Delta r_2(h', t) \Rightarrow r_1(h, t)$, where $r_1 = \text{derivationally_related_form}$ and $r_2 = \text{hypernym}$. This is a typical case where the composite relation actually equals to the first relation in the relation path. As shown in Figure 2(e), while most embedding dimensions still agree well between the actual composite relation and the calculated relation path, the distribution tends to disperse to a large range, indicating the existence of ambiguity and interaction between entities and relations. To further explore the ambiguity issue in the above case, we perform the

same analysis on another small portion of triplets that actually give $r_1(h, h')\Delta r_2(h', t) \Rightarrow r_3(h, t)$, where $r_3 = \text{synset_domain_topic_of}$ (Figure 2(f)). Interestingly, we also observe that part of embedding dimensions of $\mathcal{O}(\mathbf{r}_2)\mathcal{O}(\mathbf{r}_1)$ are aligned with $\mathcal{O}(\mathbf{r}_3)$ (embedding difference close to zero), demonstrating the flexibility of our model to capture potentially ambiguous relation compositions. On the other hand, we also observe that the model can learn to put the rotation axis of r_1 collinear with the embedding of head entities h in the composition mode: $r_1(h, h')\Delta r_2(h', t) \Rightarrow r_1(h, t)$ as shown in Figure 2(g), reflecting the interaction of entities and relations. This example clearly demonstrates the interpretability of our method in modeling complex composition patterns. The geometric patterns for other relation patterns can be found in the **Appendix**.

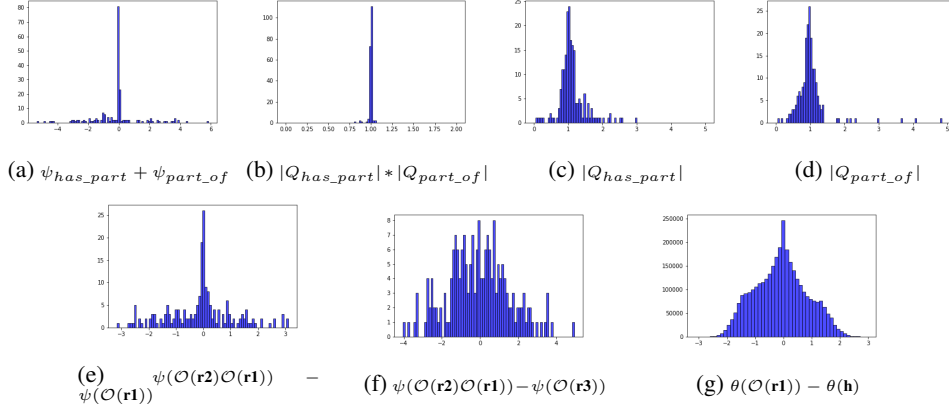


Figure 2: Geometric interpretation provided by DenseE. Each histogram shows a distribution of each dimension of the learned embeddings. Angular parameters are in radian units. (a)-(d) A case study for inversion patterns (Other degrees of freedom can be found in the **Appendix**). (e)-(f) A case study for composition patterns, reflecting the scenario of $r_1(h, h')\Delta r_2(h', t) \Rightarrow r_1(h, t)$ and $r_1(h, h')\Delta r_2(h', t) \Rightarrow r_3(h, t)$, respectively. $\psi(\cdot)$ denotes the rotation angle about the rotation axis of a relational operator. (g) Collinearity of entity and relation embedding. $\theta(\mathcal{O}(\mathbf{r}_1))$ is the θ component of relation r_1 . $\theta(\mathbf{h})$ is the θ value of the head entity h 's embedding in the spherical coordinate system of the 3-D Euclidean space. All the head entities satisfying $r_1(h, h')\Delta r_2(h', t) \Rightarrow r_1(h, t)$ are included for the analysis. The entities and relations involved can be found in the **Appendix**.

Ablation study To examine the effectiveness of each module in our model, we perform a series of ablation experiments (Table 4). On WN18 dataset, we observe no significant performance decrease except for eliminating reciprocal training, which is beneficial in modeling atomic relations. On WN18RR dataset, the most significant performance decrease occurs when we cancel the scaling operation, i.e., only model the relation as rotations. This confirms the contribution from scaling to the whole model. On FB15K-237 and YAGO3-10 datasets, we also observe a large drop of performance when removing the scaling operation. Also, self-adversarial negative sampling (adv) shows significant contribution, indicating the necessity to incorporate proper training techniques.

Table 4: Ablation study on WN18, WN18-RR, FB15K-237 and YAGO3-10 datasets.

	WN18		WN18RR		FB15K-237		YAGO3-10	
Model	MRR	H@1	MRR	H@1	MRR	H@1	MRR	H@1
DenseE	0.950	0.945	0.491	0.443	0.349	0.256	0.541	0.465
-scaling	0.950	0.945	0.475	0.432	0.335	0.240	0.486	0.392
-reciprocal	0.944	0.939	0.487	0.445	0.343	0.251	0.530	0.453
-adv	0.950	0.945	0.486	0.443	0.306	0.219	0.429	0.342

7 Conclusion

In this work, we propose an effective method, named DenseE, for knowledge graph embedding. DenseE decomposes a relation operator into an $SO(3)$ group-based rotation as well as a scaling transformation. Extensive experiments show that DenseE possesses overall state-of-the-art performance in knowledge

completion with much lower model complexity. Also, DensE provides a straightforward geometric interpretation for the relations, leading to meaningful insights for complex relation patterns.

Broader Impact

In this work, we present significant progress in reasoning capability relating to KG embedding. Harvesting knowledge is the ultimate goal that human beings seek along the advancement of technology and humanity. This research, along with many KG embedding representation studies, is attempting to design systems that have better use of human knowledge. Technically speaking, our method possesses both higher modeling performance and lower computation cost. This especially fits the current trend in applying knowledge graph on a larger and larger scale and the initiative on energy saving in the AI application.

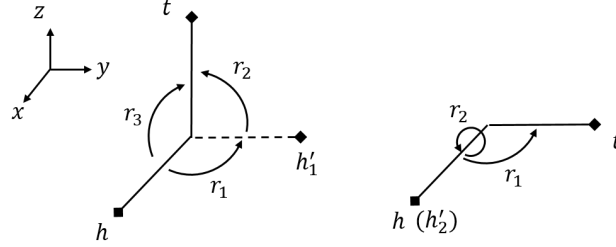
When it comes to who may be put at disadvantage from this research, we do not think it is applicable since our study of addressing the reasoning capabilities of AI is still at an early stage. Having said so, we are fully aware that we need to develop AI ethically and responsibly. The potential bias in the data that may lead to wrong knowledge needs to be cleaned or corrected with validation and interpretation mechanisms. Therefore, in this study, we are devoted to bringing interpretability to the reasoning process. In the future, we will continue to put transparency of modeling knowledge composition as a focus in our study.

Appendix

A Source code

We have published DenseE as an open source software and the source code can be found online at <https://github.com/anonymous-dense-submission/DenseE>

B The geometrical interpretation of non-commutative compositions



$$r_1(h, h'_1) \wedge r_2(h'_1, t) \Rightarrow r_3(h, t) \quad r_2(h, h'_2) \wedge r_1(h'_2, t) \Rightarrow r_1(h, t)$$

Figure 3: A simple example for how rotation in the 3-D Euclidean space can model non-commutative relations

Left: A rotation about axis- z followed by a rotation about axis- x , the initial vector \mathbf{h} is placed along axis- x . It can be seen that the final state is along axis- z , and two rotation operations are equivalent to one operation with rotation axis to be about axis- y .

Right: The rotation operation sequence is reversed from the left figure. A rotation about axis- x is followed by a rotation about axis- z . The final state is then changed to be along axis- y . Since the initial vector is collinear with the first rotation axis, the two rotation operations are equal to the last rotation (rotation about axis- z).

C Formal definitions and examples of relation patterns

Here, we formally define the relation patterns discussed in our work, each of which is followed by an example for intuitive understanding. Let x, y, z be the entities in a given KG, and $r(\cdot, \cdot)$ maps the relation between the two entities, we have:

Definition 1. Relation r_3 is **composed** of relation r_1 and relation r_2 if $\forall x, y, z$

$$r_1(x, y) \wedge r_2(y, z) \Rightarrow r_3(x, z) \quad (8)$$

Here r_3 is also referred to as a **composite relation** and possesses certain **composition pattern**. A typical example is: given y is x 's father and z is y 's wife, we can infer the relationship between x and z .

Definition 2. A relation r is **symmetric/anti-symmetric** if $\forall x, y$

$$r(x, y) \Rightarrow r(y, x) \quad (r(x, y) \Rightarrow \neg r(y, x)) \quad (9)$$

Friend is a typical example of symmetric relation, which means if we know x is friend of y , we can infer y is also friend of x . *Filiation* is an example of anti-symmetric relation.

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

$$r_1(x, y) \Rightarrow r_2(y, x) \quad (10)$$

For instance, *has_part* and *part_of* fit into the scope of inverse relations, which means if we know x is a part of y , we can infer that y has part x . Note that both **symmetric/antisymmetric** and **inverse** relation patterns can be inferred in one hop, so they are also called **atomic** relation.

Table 5: Statistics of datasets used in this study

Dataset	# Entities	# Relations	#Training	#Validation	#Test
WN18	40943	18	141442	5000	5000
WN18RR	40943	11	86835	3034	3134
FB15k-237	14541	237	272115	17535	20466
YAGO3-10	123182	37	1079040	5000	5000

D The basic statistics of the benchmark datasets

The experiments are conducted on four commonly used benchmark datasets, including WN18, WN18RR, FB15k-237 and YAGO3-10. WN18 comes from WordNet [8], which consists of many lexical relations of English words. As WN18 has a significant part of inverse relations, it is mainly used here to reflect inversion pattern of relations. A new dataset, named WN18RR [2], has been proposed where inverse relations are removed. Similarly, the FB15k-237 dataset [14] is extracted from the original Freebase dataset FB15K [1] by removing inverse relations. In addition, we also use the YAGO3-10 [7] data, which consists of a large collection of triplets from multilingual Wikipedia. The latter three datasets are involved to assess the model performance on composition patterns. The basic statistics of these datasets are provided in Supplementary Table 5.

E Link prediction results with more evaluation metrics

In the main text, we report Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hits at 1 (H@1) for evaluation for each dataset. Here, we provide more metrics including Mean Rank (MR), H@3 and H@10. The results are shown in Supplementary Table 6 and Supplementary Table 7.

Table 6: Performance comparison on WN18 and WN18-RR datasets. Best results are labeled in bold and the second best are underlined. Performance of TransE is from [11] and [10]; Performance of ComplEx is from [2]; Performance of RotatE, QuatE and NagE are taken from the corresponding original papers. QuatE and QuatE⁺ denote the plain QuatE model and QuatE model with N3 regularization and reciprocal learning, respectively.

Model	WN18					WN18RR				
	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE	-	0.495	0.113	0.888	0.943	3384	0.226	-	-	0.501
DistMult	655	0.797	-	-	0.946	5110	0.43	0.39	0.44	0.49
ComplEx	-	0.941	0.936	0.945	0.947	5261	0.44	0.41	0.46	0.51
RotatE	<u>309</u>	0.949	<u>0.944</u>	0.952	0.959	<u>3340</u>	0.476	0.428	0.492	0.571
NagE	-	0.950	<u>0.944</u>	<u>0.953</u>	0.960	-	0.477	0.432	0.493	<u>0.574</u>
QuatE	388	<u>0.949</u>	0.941	0.954	<u>0.960</u>	3472	0.481	<u>0.436</u>	0.500	0.564
QuatE ⁺	-	0.950	<u>0.944</u>	0.954	0.962	-	<u>0.482</u>	<u>0.436</u>	<u>0.499</u>	0.572
DensE	285	0.950	0.945	0.954	0.959	3052	0.491	0.443	0.508	0.579

F Hyperparameters setting

The ranges of the hyperparameters for the grid search are set as follows: Embedding size $k \in \{100, 200, 500, 1000\}$ (In our model, each entity is represented with a matrix with a size of $3 \times k$, and each relation with a matrix with a size of $4 \times k$), batch size $b \in \{256, 512, 1024\}$, fixed margin $\gamma \in \{3.0, 6.0, 9.0, 12.0, 15.0, 24.0, 30.0\}$, negative sampling size $n \in \{256, 512, 1024\}$, self-adversarial sampling temperature $\alpha \in \{0.3, 0.5, 1.0\}$. The initial learning rate η is set to be 0.1, and it decays with a factor of 1/2 if the training loss does not decrease in 1000 epochs. We list the best hyperparameters setting of DensE on the benchmark datasets in Supplementary Table 8.

Table 7: Performance comparison on FB15k-237 and YAGO3-10 datasets. Best results are labeled in bold and the second best are underlined. Performance of TransE is from [10]; Performance of ComplEx is from [2]; Performance of RotatE, QuatE and NagE are taken from the corresponding original papers. QuatE and QuatE⁺ denote the plain QuatE model and QuatE model with N3 regularization and reciprocal learning, respectively.

Model	FB15k-237					YAGO3-10				
	MR	MRR	H@1	H@3	H@10	MR	MRR	H@1	H@3	H@10
TransE	357	0.294	-	-	0.465	-	-	-	-	-
DistMult	254	0.241	0.155	0.263	0.419	5926	0.34	0.24	0.38	0.54
ComplEx	339	0.247	0.158	0.275	0.428	6351	0.36	0.26	0.4	0.55
RotatE	177	0.338	0.241	0.375	0.533	<u>1767</u>	<u>0.495</u>	<u>0.402</u>	<u>0.55</u>	<u>0.67</u>
NagE	-	0.340	0.244	0.378	0.530	-	-	-	-	-
QuatE	<u>176</u>	0.311	0.221	0.342	0.495	-	-	-	-	-
QuatE ⁺	-	0.366	0.271	0.401	0.556	-	-	-	-	-
DensE	169	<u>0.349</u>	<u>0.256</u>	<u>0.384</u>	<u>0.535</u>	1450	0.541	0.465	0.585	0.678

Table 8: Hyperparameters settings of DensE in this study.

Dataset	Embedding size k	Batch size b	Margin γ	Negative sample size n	adv temperature α
WN18	200	512	12.0	1024	0.3
WN18RR	200	512	6.0	512	0.5
FB15k-237	500	1024	9.0	256	1.0
YAGO3-10	200	1024	24.0	512	1.0

G Capability of DensE in modeling relation patterns

In this Section, we provide a detailed analysis on how our method tend to model each relation pattern in an interpretable way. In our experiment, we calculate the statistical rule of each degree of freedom to reflect the effect of specific relation patterns. In addition, we also sometimes compare the embeddings of two relation types (or a relation type and a entity) *per element*, i.e., we perform element-wise addition, subtraction, multiplication on each embedding dimension. Then, we use the distribution of these results to demonstrate how the two compared embeddings agree with each other. Note that below we use an addition subscript i to denote each dimension in the embeddings.

G.1 Symmetry/anti-symmetry pattern

As pointed out in main text **Section 4.4**, for the **symmetry** relation pattern, the scaling factor $|Q|$ of symmetric relation tend to be one, and the rotation angle ψ should be 0 or π in $[0, 2\pi)$. For **anti-symmetry** relation pattern, one can easily check that the scaling factor $|Q|$ should also be one, but the rotation angle ψ should be neither 0 nor π in the range of $[0, 2\pi)$. Here we show the distributions of rotation angle ψ and scaling factor $|Q|$ of four relations with **symmetry** pattern in WN18RR (Supplementary Figure 4). It should be noted that since the embedding size k in our model for WN18RR is set to be 200, the sum of frequency in these distributions also equals to 200.

G.2 Inversion pattern

In main text **Section 4.4**, we assert that if two relations r_1 and r_2 satisfy the **inverse** pattern, if and only if they satisfy: $|\mathbf{r}_{1i}| * |\mathbf{r}_{2i}| = 1$, $\theta_{r_{1i}} = \theta_{r_{2i}}$, $\phi_{r_{1i}} = \phi_{r_{2i}}$ and $\psi_{r_{1i}} = -\psi_{r_{2i}}$. In Supplementary Figure 5, we show a case of paired relations with **inversion** pattern from WN18 dataset, namely *has_part, part_of*. We plot the scaling factor $|Q|$, magnitude of the rotation ψ , and (θ, ϕ) that describe the rotation axis for each relation (first two columns), as well as their element-wise alignment results (the last column).

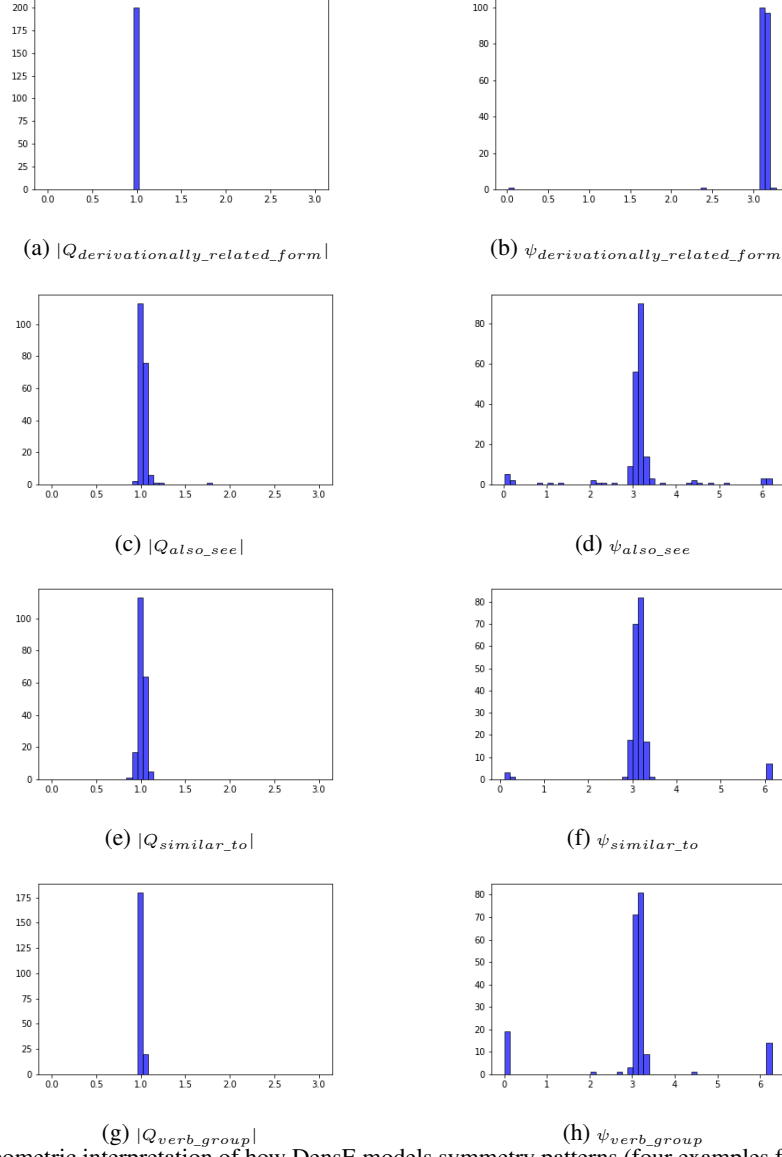


Figure 4: Geometric interpretation of how DenseE models symmetry patterns (four examples from WN18RR). Each row shows the distribution of $|Q|$ and ψ for a given relation, respectively.

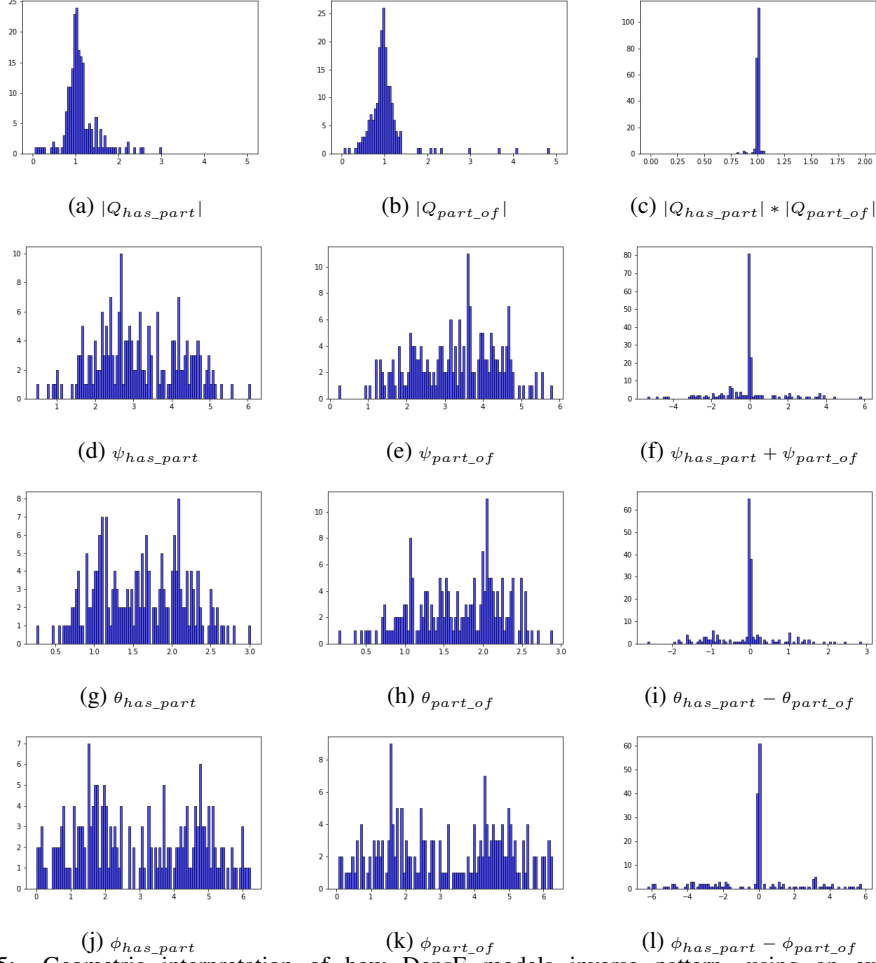


Figure 5: Geometric interpretation of how DensE models inverse pattern, using an example of $(has_part, part_of)$ from WN18. At each row, we show the embedding from one degree of freedom of our model. The first two columns show the embeddings of each relation type, and the last column shows the alignment of the two embeddings regarding a specific degree of freedom.

G.3 Composition pattern

G.3.1 Ambiguity in composition pattern

In a real-world KG (here we take a sub-graph from WN18RR as an example), due to the ambiguity issue mentioned in the main text (composition pattern **Property 2**), there exist plenty of examples where a third relation (the composite relation) cannot be inferred given the two participating relations alone (Supplementary Figure 6(a)). For example, given $r_1 = \text{derivationally_related_form}$ and $r_2 = \text{hypernym}$, we have the composition pattern as shown with the blue lines: (Trade(VB), derivationally_related_form, Trade(NN)), (Trade(NN), hypernym, transaction) and (Trade(VB), derivationally_related_form, transaction). From these cases, it seems that one can summarize the composition pattern as: $r_3 = r_1 = \text{derivationally_related_form}$, i.e., $r_1(h, h') \wedge r_2(h', t) \Rightarrow r_1(h, t)$. However, we also have the triangle with red lines, i.e., (Trade(VB), derivationally_related_form, Selling), (Selling, hypernym, mercantilism) and (Trade(VB), synset_domain_topic_of, mercantilism). In these cases, it looks like the composition pattern has the form that $r_1(h, h') \wedge r_2(h', t) \Rightarrow r_3(h, t)$, where $r_3 = \text{synset_domain_topic_of}$. This ambiguity means that the composition mode is not uniform but depends on specific entities and their other neighborhoods. Therefore, in order to give the model sufficient flexibility to learn this, our model does not require all the dimensions in a relation embedding to fit in one single composition mode (e.g., $r_1(h, h') \wedge r_2(h', t) \Rightarrow r_1(h, t)$ or $r_1(h, h') \wedge r_2(h', t) \Rightarrow r_3(h, t)$). In consequence, the learned relation embedding for a com-

posite relation are actually distributed in a disperse manner, with the majority of embedding dimensions following mode $r_1(h, h')\Delta r_2(h', t) \Rightarrow r_1(h, t)$, and some minor portions following $r_1(h, h')\Delta r_2(h', t) \Rightarrow r_3(h, t)$, which is consistent with the abundance of each mode in the training data.

G.3.2 Case study: a two-hop composition pattern

As we mentioned in the main text, in a composition pattern, the relations involved are not necessarily different (composition pattern **Property 3**). Here we discuss the simplest case of a two-hop relation composition. Even under the assumption that the composite relation can be inferred from its composition alone (i.e., an **unambiguous** composition), there are still five possible situations, as shown in Supplementary Figure 6(b).

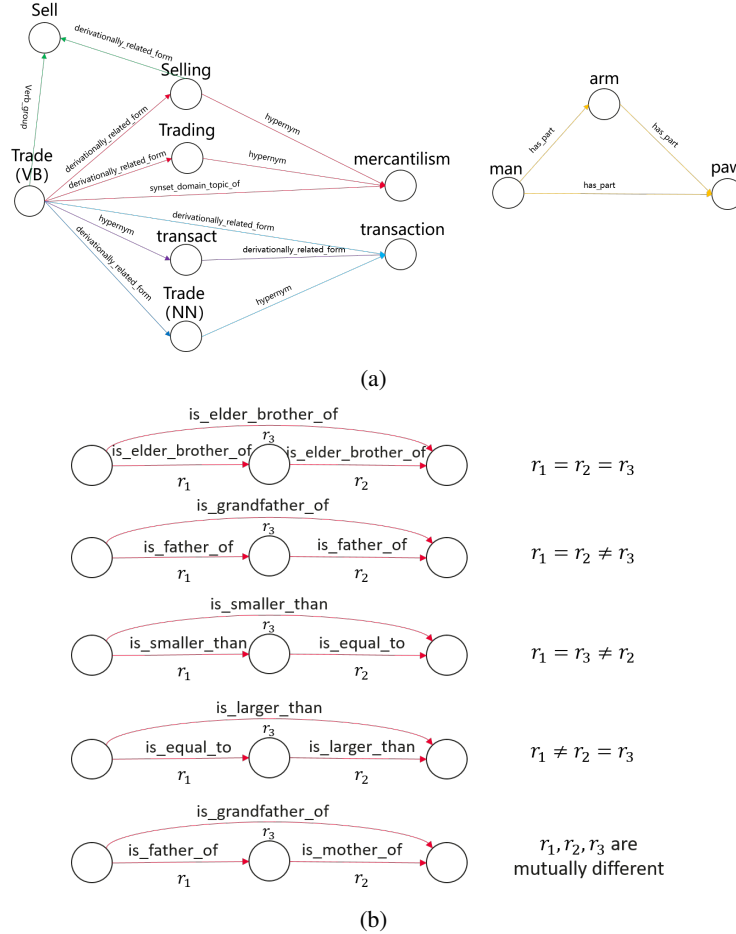


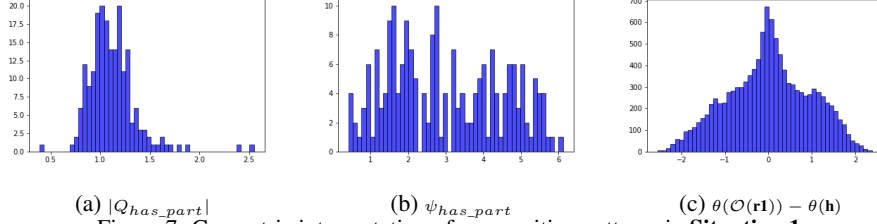
Figure 6: (a) A representative subgraph from WN18RR to show ambiguity. (b) The relations involved in a composition pattern are not necessarily to be mutually different.

Here we provide a detailed analysis on these five situations, using real cases in WN18RR as examples (Supplementary Figure 6(a)).

Situation 1: r_1, r_2 and r_3 are all the same, e.g., the triangle in yellow color with three triplets: (man, has_part, arm) , (arm, has_part, paw) and (man, has_part, paw) . To satisfy the composition relation constraint, in principle the model can adopt several approaches to fit this situation. From the perspective of relation embedding, it can learn to fit $|Q_{has_part}| = 1$ or $\psi_{has_part} \in \{0, 2\pi\}$. Also, in terms of interaction between entities and relations, it can also learn to align the rotation axis of relation with the entity embedding (see Supplementary Figure 7(c), showing the differences between the θ component of r_1 in the mode $r_1(h, h')\Delta r_1(h', t) \Rightarrow r_1(h, t)$ and the corresponding head entities h in spherical coordinate system). Note that when the rotation axis aligns with head

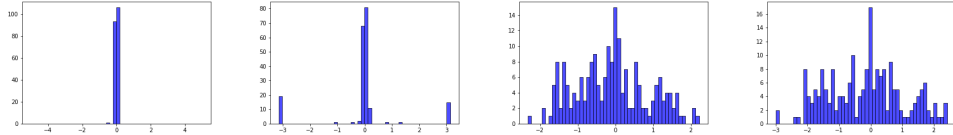
entity embedding in spherical coordinate system, the composition pattern can be modeled by the scaling factor alone, i.e., $|Q_{has_part}|^2 = |Q_{has_part}|$, regardless of the rotation magnitude ψ .

In this case, we see the model mainly adopts the first approach (Supplementary Figure 7(a)), as $|Q_{has_part}|$ is roughly around 1. In addition, the partial alignment between relation rotation axis and entities is also observed (Supplementary Figure 7(c)). In contrast, for ψ_{has_part} , we see very few dimensions satisfy the condition (0 or 2π), which is also consistent with the dominate role of scaling factor in this case.



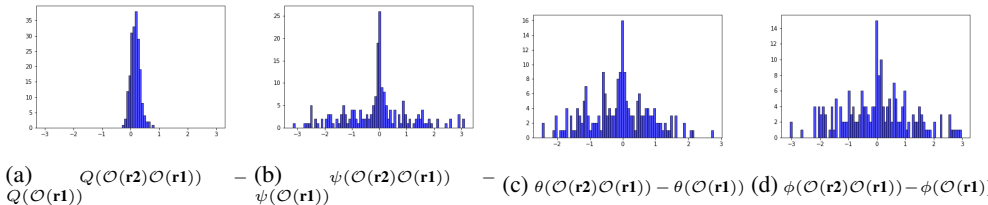
(a) $|Q_{has_part}|$ (b) ψ_{has_part} (c) $\theta(\mathcal{O}(\mathbf{r1})) - \theta(\mathbf{h})$
Figure 7: Geometric interpretation of composition patterns in **Situation 1**.

Situation 2: r_1 and r_2 are the same, but not equal to r_3 . An example is given by the triangle in green with three triplets: (*Trade*(VB), *derivationally_related_form*, *Selling*), (*Selling*, *derivationally_related_form*, *Sell*) and (*Trade*(VB), *verb_group*, *Sell*). As mentioned in main text **Section 4.4**, we expect the model to learn $\psi_{verb_group} = 2\psi_{derivationally_related_form}$ and $|Q_{verb_group}| = |Q_{derivationally_related_form}|^2$, both of which are confirmed in this case. Besides this, we also observe that the rotation axes of the two relations are aligned on some dimensions, i.e., $\theta_{verb_group} = \theta_{derivationally_related_form}$ and $\phi_{verb_group} = \phi_{derivationally_related_form}$. The corresponding distributions are shown in Supplementary Figure 8, where we have $r_1 = r_2 = derivationally_related_form$ and $r_3 = verb_group$.



(a) $Q(\mathcal{O}(\mathbf{r3})) - Q^2(\mathcal{O}(\mathbf{r1}))$ (b) $\psi(\mathcal{O}(\mathbf{r3})) - 2 * \psi(\mathcal{O}(\mathbf{r1}))$ (c) $\theta(\mathcal{O}(\mathbf{r3})) - \theta(\mathcal{O}(\mathbf{r1}))$ (d) $\phi(\mathcal{O}(\mathbf{r3})) - \phi(\mathcal{O}(\mathbf{r1}))$
Figure 8: Geometric interpretation of composition patterns in **Situation 2**.

Situation 3: r_1 and r_3 are the same, but not equal to r_2 , e.g., the triangle in blue color with three triplets set: (*Trade*(VB), *derivationally_related_form*, *Trade*(NN)), (*Trade*(NN), *hypernym*, *transaction*) and (*Trade*(VB), *derivationally_related_form*, *transaction*). Here we compare the difference between the embedding of a composite relation and the embedding calculated by multiplying each relation in the relation path. We have already shown the distribution of ψ in the main context. Other distributions are shown in Supplementary Figure 9, where we have $r_1 = r_3 = derivationally_related_form$ and $r_2 = hypernym$.



(a) $Q(\mathcal{O}(\mathbf{r2})\mathcal{O}(\mathbf{r1})) - Q(\mathcal{O}(\mathbf{r1}))$ (b) $\psi(\mathcal{O}(\mathbf{r2})\mathcal{O}(\mathbf{r1})) - \psi(\mathcal{O}(\mathbf{r1}))$ (c) $\theta(\mathcal{O}(\mathbf{r2})\mathcal{O}(\mathbf{r1})) - \theta(\mathcal{O}(\mathbf{r1}))$ (d) $\phi(\mathcal{O}(\mathbf{r2})\mathcal{O}(\mathbf{r1})) - \phi(\mathcal{O}(\mathbf{r1}))$
Figure 9: Geometric interpretation of composition patterns in **Situation 3**.

Situation 4: r_2 and r_3 are the same, but not equal to r_1 , e.g., the triangle in purple color with three triplets set: (*Trade*(VB), *hypernym*, *transact*), (*transact*, *derivationally_related_form*, *transaction*) and (*Trade*(VB), *derivationally_related_form*, *transaction*). Here we compare the difference between the embedding of a composite relation and the embedding calculated by

Table 9: The mean values and variance of MRR on WN18, WN18RR, FB15k-237 and YAGO3-10 datasets.

Dataset	WN18	WN18RR	FB15k-237	YAGO3-10
MRR	0.950±0.001	0.491±0.001	0.349±0.001	0.541±0.001

multiplying each relation in the relation path. Related distributions are shown in Supplementary Figure 10, where we have $r_1 = \text{hypernym}$ and $r_2 = r_3 = \text{derivationally_related_form}$. In this case, we see that the composition pattern given by $\mathcal{O}(\mathbf{r}_2)\mathcal{O}(\mathbf{r}_1)$ is learned to have similar scaling factor $|Q|$ and rotation magnitude ψ with r_2 , and the rotation axes (θ, ϕ) are also partially aligned.

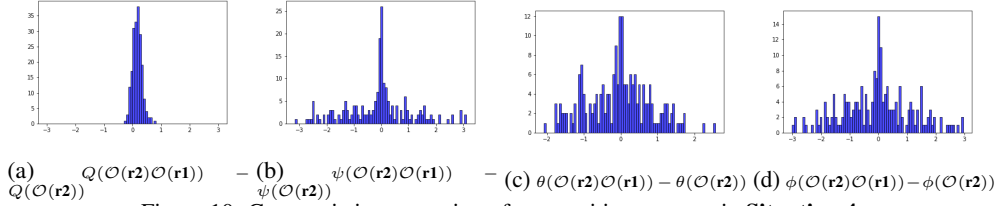


Figure 10: Geometric interpretation of composition patterns in **Situation 4**.

Situation 5: r_1 , r_2 and r_3 are mutually different relations, e.g., the triangle in red color with three triplets set: $(\text{Trade}(\text{VB}), \text{derivationally_related_form}, \text{Selling})$, $(\text{Selling}, \text{hypernym}, \text{mercantilism})$ and $(\text{Trade}(\text{VB}), \text{synset_domain_topic_of}, \text{mercantilism})$. Here we compare the difference between the embedding of a composite relation and the embedding calculated by multiplying each relation in the relation path. Related distributions are shown in Supplementary Figure 11, where we have $r_1 = \text{derivationally_related_form}$, $r_2 = \text{hypernym}$ and $r_3 = \text{synset_domain_topic_of}$. The reason for relatively large dispersion here is discussed in the main text, i.e., the majority of triplets exemplify $\text{derivationally_related_form} \wedge \text{hypernym} \Rightarrow \text{derivationally_related_form}$ (See also the discussion above of **Situation 3**), while only a small portion have the pattern $\text{derivationally_related_form} \wedge \text{hypernym} \Rightarrow \text{synset_domain_topic_of}$.

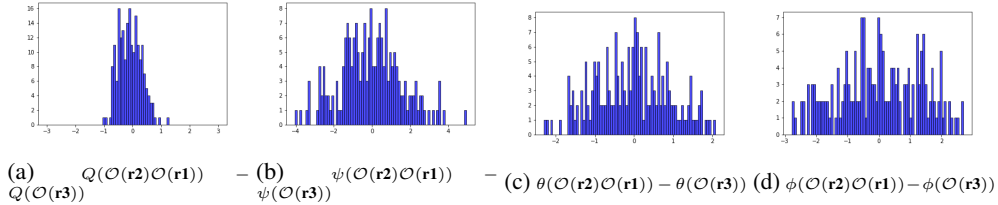


Figure 11: Geometric interpretation of composition patterns in **Situation 5**.

H Variance of the prediction performance

The mean values and corresponding variance of MRR on WN18, WN18RR, FB15k-237 and YAGO3-10 datasets are shown in Supplementary Table 9. The results are obtained by training DensE with five different random seeds, showing that the prediction performance of DensE is quite stable.

I Effect of self-adversarial negative sampling on DensE and RotatE

In the ablation study, we observe a significant contribution of the self-adversarial negative sampling technique on the prediction performance of FB15k-237 and YAGO3-10. Therefore, we compare our model with RotatE in the setting where both models are trained without self-adversarial negative sampling (Supplementary Table 10). These results further confirm the superiority of our model without self-adversarial negative sampling.

Table 10: Results of DensE and RotatE without self-adversarial negative sampling training technique, where “adv” represents “self-adversarial”.

Model	WN18	WN18RR	FB15k-237	YAGO3-10
DensE (w/o adv)	0.950	0.486	0.306	0.429
RotatE (w/o adv)	0.947	0.470	0.297	-

References

- [1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.
- [2] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [3] Katsuhiko Hayashi and Masashi Shimbo. A non-commutative bilinear model for answering path queries in knowledge graphs. In *EMNLP/IJCNLP*, 2019.
- [4] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020.
- [5] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *ICML*, 2018.
- [6] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP/IJCNLP*, 2019.
- [7] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. Yago3: A knowledge base from multilingual wikipedias. 2013.
- [8] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [9] Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, 2019.
- [10] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121*, 2017.
- [11] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Thirtieth Aaii conference on artificial intelligence*, 2016.
- [12] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.
- [13] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *The Seventh International Conference on Learning Representations*, 2019.
- [14] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, 2015.
- [15] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.
- [16] An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy, July 2019. Association for Computational Linguistics.

- [17] Tong Yang, Long Sha, and Pengyu Hong. A group-theoretic framework for knowledge graph embedding. *ArXiv*, abs/2005.10956, 2020.
- [18] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, pages 2731–2741, 2019.
- [19] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *ArXiv*, abs/1905.07129, 2019.