NagE: Non-Abelian Group Embedding for Knowledge Graphs

Tong Yang*
Department of Physics,
Boston College, Chestnut Hill,
Massachusetts, USA

Long Sha*
Department of Computer Science,
Brandeis University, Waltham,
Massachusetts, USA

Pengyu Hong
Department of Computer Science,
Brandeis University, Waltham,
Massachusetts, USA

ABSTRACT

We demonstrated the existence of a group algebraic structure hidden in relational knowledge embedding problems, which suggests that a group-based embedding framework is essential for designing embedding models. Our theoretical analysis explores merely the intrinsic property of the embedding problem itself hence is model independent. Motivated by the theoretical analysis, we have proposed a group theory-based knowledge graph embedding framework, in which relations are embedded as group elements, and entities are represented by vectors in group action spaces. We provide a generic recipe to construct embedding models associated with two instantiating examples: **SO3E** and **SU2E**, both of which apply a continuous non-Abelian group as the relation embedding. Empirical experiments using these two exampling models have shown state-of-the-art results on benchmark datasets.

CCS CONCEPTS

• **Computing methodologies** → **Knowledge representation and reasoning**; *Statistical relational learning*.

ACM Reference Format:
Tong Yang, Long Sha, and Pengyu Hong. 2020. NagE: Non-Abelian Group Embedding for Knowledge Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3340531.3411875

1 INTRODUCTION

Knowledge graphs (KGs) are prominent structured knowledge bases for many downstream semantic tasks [7]. A KG contains an entity set $\mathcal{E} = \{e_i\}$, which correspond to vertices in the graph, and a relation set $\mathcal{R} = \{r_k\}$, which forms edges. The entity and relation sets form a collection of factual triplets, each of which has the form $(\mathbf{e}_i, \mathbf{r}_k, \mathbf{e}_j)$ where $\mathbf{r}_k$ is the relation between the head entity $\mathbf{e}_i$ and the tail entity $\mathbf{e}_j$. Since large scale KGs are usually incomplete due to missing links (relations) amongst entities, an increasing amount of recent works [2, 9, 15, 17] have devoted to the graph completion (i.e., link prediction) problem by exploring a low-dimensional representation of entities and relations.

*Equal Contribution

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411875

More formally, each relation $\mathbf{r}$ acts as a mapping $\mathbf{O}_\mathbf{r}[\cdot]$ from its head entity $\mathbf{e}_1$ to its tail entity $\mathbf{e}_2$:

$$\mathbf{r} : \mathbf{e}_1 \mapsto \mathbf{O}_\mathbf{r}[\mathbf{e}_1] =: \mathbf{e}_2. \tag{1}$$

The original KG dataset represents these mappings in a tabular form, and the task of knowledge graph embedding (KGE) is to find a better representation for these abstract mappings. For example, in the TransE model [2], relations and entities are embedded in the same vector space, and the operation $\mathbf{O}_\mathbf{r}[\cdot]$ is simply a vector summation: $\mathbf{O}_\mathbf{r}[\mathbf{e}] = \mathbf{e} + \mathbf{r}$. In general, the operation could be either linear or nonlinear, either pre-defined or learned. Importantly, graph completion relies on the fact that relations are not independent. For example, the hypernym and hyponym are inverse to each other; while kinship relations usually support mutual inferences. These dependencies would impose constraints on the operation design. Previous studies [13, 16] have concerned some specific cases of inter-relation dependencies, including (anti-)symmetry and compositional relations.

In this work, however, we attempt to deliver a high-level analysis from a general perspective. More particularly, we ask the following three questions:

(1) What constraints/requirements does a general KGE task impose on embeddings?
(2) What kind of embedding method would satisfy these constraints?
(3) How to explicitly construct embedding models?

Note that the first question concerns general KGE tasks rather than specific datasets nor embedding models, and, therefore, requires an analysis including all possible knowledge graph structures. We find that, to accommodate all possible KG datasets, there are five requirements for the relation embedding: ***closure, identity, inverse, associativity,*** and ***non-commutativity***. The first four coincide with the algebraic definition of groups in mathematics, and imply a direct answer to the second question: embedding all relations into a group manifold (and designing mapping operations as group actions) would automatically satisfy all requirements; in addition, the last requirement, non-commutativity, further suggests implementing non-Abelian groups for the most general KGE tasks. The third question asks for a general recipe to embed relations as group elements.

One main contribution of this work is it provides a framework for addressing the KG embedding problem from a novel and more rigorous perspective: the group-theoretic perspective. We prove that the intrinsic structure of general KGE tasks coincides with the complete definition of groups. To our best knowledge, this is the first proof that rigorously legitimates the application of group theory in KG embedding. With this framework, we also establish connections with many existing models (see Sec 3.3), including: TransE [2], TransR [9], TorusE [5], RotatE [13], ComplEx [15], DisMult [17].

The remaining sections are organized as following: Section 2 mentions some related works, and emphasizes the distinction between our analysis and others; in Section 3, we answer the first two question proposed above, achieving a conclusion that continuous non-Abelian groups suit general KGE tasks well, which leads to the exampling continuous non-Abelian group embedding models (**NagE**) in later sections; in Section 4, we provide a general recipe for group-embedding approach of KGE problems, associated with two novel instantiating models: **SO3E** and **SU2E**, which completes the answer for all three questions; we demonstrate the power of the proposed embedding framework by comparing the experimental results of our two example models with other state-of-the-art models in Section 5, and conclude all discussions in Section 6.

2 RELATED WORKS

From the group theory perspective, our work may be related to the TorusE [5], the RotatE [13], DihEdral [16], and QuatE [18] models.

The TorusE model frames the KG entity embedding in a Lie group manifold to deal with the *compactness problem*. The authors proved that the additive nature of the operation in the TransE model contradicts the entity regularization. However, if a non-compact embedding space is used, entities must be regularized to prevent the divergence of negative scores. Therefore the TorusE model used n-torus, a compact manifold, as the entity embedding space. In other words, TorusE embeds entities in group manifolds, while our work embeds relations as group manifolds.

In the DihEdral model, the D_4 group the author used plays the same role as groups in our work: group elements correspond to relations. The motivation of DihEdral is to resolve the non-Abelian composition (i.e., the compositional relation formed by $\mathbf{r}_1$ and $\mathbf{r}_2$ would change if the two are switched). Nevertheless, DihEdral applies a discrete group D_8 for relation embedding while using a continuous entity embedding space, which may suffer two problems as discussed in the later Section 3.3. The RotatE model was designed to accommodate symmetric, inversion, and (abelian) composition of triplets at the same time. Different from these previous works, our work does not target at one or few specific cases but aims at answering the more general question: finding the generative principle of all possible cases, and thus to provide guidance for model designs that can accommodate all cases.

More importantly, in most preceding works related to groups [3, 16], group theory serves as an alternative perspective to explain the efficiency of specific models; while the theoretical analysis in our current work in Section 3 is model/dataset independent and is merely initiated by KGE tasks themselves, and the group embedding approach automatically emerges as the natural method which could satisfy all constraints for a general KGE task. To our the best of our knowledge, this is the first proof that rigorously legitimates the implementation of group theory in KG embeddings.

Another interesting connection refers to QuatE [18] model, where the authors proposed quaternions (also octonions and sedenions in the appendix), which served as an extension of complex numbers for KGE. The intuition of their model was not related to group theory at all. However, in Model Analysis (Section 5.3 in [4]), the authors stated that: "Normalizing the relation to unit quaternion is a critical step for the embedding performance." While this was empirically observed, an explicit reason is absent. This phenomena can be easily understood from the group theory perspective, as long as one realizes the mathematical correspondence: SU(2) group is an isomorphism to unit quaternions. This explains the necessity of applying "normalization" on quaternions: only unit quaternions are consistent with the group structure (SU(2) in specific), while non-unit ones cannot form a group. One of our newly proposed model, **SU2E**, is therefore closely related to QuatE model with "unit-scaling", although our proposal does not concern different number systems at all. It is also worth to mention that when comparing with other proceeding models, the authors applied a completely different criteria for the QuatE experiments: a "type constraint" was introduced in their experiments, which filtered out a significant portion of challenging relation types during the evaluation phase. As a contrast, our **SU2E** model proposed in later sections investigated the similar setting but compared with other models under the common criteria (without "type constraint"), and showed superior results as a *continuous non-Abelian group embedding* for the first time.

3 GROUP THEORY IN RELATIONAL EMBEDDINGS

In this section, we formulate the group-theoretic analysis for relational embedding problems. Firstly, as most embedding models represent objects (including entities and relations) as vectors, the task of operation design thus can be stated as finding proper transformations of vectors. Secondly, as we mentioned in the introduction, our ultimate goal of reproducing the whole knowledge graph using atomic triplets further requires certain types of local patterns to be accommodated. We now discuss these structures, which in the end naturally leads to the definition of groups in mathematics.

3.1 Hyper-relation Patterns: relation-of-relations

One difficulty of generating the whole knowledge graph from atomic triplets lies in the fact that different relations are not independent of each other. The task of relation inference relies exactly on their mutual dependency. In other words, there exist certain *relation of relations* in the graph structure, which we term as *hyper-relation patterns*. A proper relation embedding method and the associated operations should be able to capture these hyper-relations.

Now instead of studying exampling cases one by one, we ask the most general question: what are the most fundamental hyper-relations? The answer is quite simple and only contains two types, namely, *inversion* and *composition*:

- **Inversion**: given a relation $\mathbf{r}$, there *may exist* an inversion $\bar{\mathbf{r}}$, such that, $\forall \mathbf{e}_1, \mathbf{e}_2 \in \mathcal{E}$:

$$\mathbf{r} : \mathbf{e}_1 \mapsto \mathbf{e}_2 \quad \longrightarrow \quad \bar{\mathbf{r}} : \mathbf{e}_2 \mapsto \mathbf{e}_1. \tag{2}$$

  The inversion captures any relation path with a length equal to 1 (in the unit of relations).
- **Composition**: given two relations $\mathbf{r}_1$ and $\mathbf{r}_2$, there *may exist* a third relation $\mathbf{r}_3$, such that, $\forall \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathcal{E}$:

$$\begin{cases} \mathbf{r}_1 : \mathbf{e}_1 \mapsto \mathbf{e}_2 \\ \mathbf{r}_2 : \mathbf{e}_2 \mapsto \mathbf{e}_3 \end{cases} \quad \longrightarrow \quad \mathbf{r}_3 : \mathbf{e}_1 \mapsto \mathbf{e}_3. \tag{3}$$

Any relation paths longer than 1 can be captured by a sequence of compositions.

One may notice the phrase *may exist* in the above definition, this simply emphasizes that the existence of these derived conceptual relations $\bar{\mathbf{r}}$ and $\mathbf{r}_3$ depends on the **specific KG dataset**; while, on the other hand, to accommodate general KG datasets, the **embedding space** should always contains the mathematical representations of these conceptual relations.

An important feature of KG is that with the above two hyper-relations, one could generate any local graph pattern and eventually the whole graph, as relational paths with arbitrary length have been captured. Note the term of *inversion* and *composition* might have different meanings from ones in other works: most existing works study triplets to analyze hyper relations, while the definition we provide above is based purely on relations. This is more general in the sense that any conclusion derived would not depend on entities at all, and some different hyper relations could, therefore, be summarized as a single one. For example, there are enormous discussions on *symmetric* triplets and *anti-symmetric* triplets [13], which are defined as:

$$\begin{aligned} \textbf{symmetric}: &\quad (\mathbf{e}_1, \mathbf{r}, \mathbf{e}_2) \quad \longrightarrow \quad (\mathbf{e}_2, \mathbf{r}, \mathbf{e}_1), \\ \textbf{anti-symmetric}: &\quad (\mathbf{e}_1, \mathbf{r}, \mathbf{e}_2) \quad \longrightarrow \quad \neg(\mathbf{e}_2, \mathbf{r}, \mathbf{e}_1). \end{aligned}$$

In fact, if for any choice of $\mathbf{e}_{1,2}$, one could produce a symmetric pair of true triplets using $\mathbf{r}$, this would imply a property of $\mathbf{r}$ itself, and in which case, one could then simply derive:

$$\bar{\mathbf{r}} = \mathbf{r}. \tag{4}$$

This is a special case of the inversion hyper-relation; and similarly, the anti-symmetric case simply implies $\bar{\mathbf{r}} \neq \mathbf{r}$, which is quite common, and does not require extra design. The deep reason for discussing hyper-relations which relies merely on relations rather than triplets is that the logic of relation inference problem itself is not entity-dependent.

3.2 Emergent Group Theory

To accommodate both general inversions and general compositions, we now derive explicit requirements on the relation embedding model. We start by defining the *product* of two relations $\mathbf{r}_1$ and $\mathbf{r}_2$: $\mathbf{r}_1 \cdot \mathbf{r}_2$, as subsequently "finding the tail" twice according to the two relations, i.e.

$$\mathbf{O}_{\mathbf{r}_1 \cdot \mathbf{r}_2}\big[\,\cdot\,\big] := \mathbf{O}_{\mathbf{r}_1}\big[\mathbf{O}_{\mathbf{r}_2}[\cdot]\big]. \tag{5}$$

With the above definition, (3) can be rewritten as: $\mathbf{r}_3 = \mathbf{r}_1 \cdot \mathbf{r}_2$. One would realize that the following properties should be supported by a proper embedding model:

(1) **Inverse element**: to allow the possible existence of inversion, the elements $\bar{\mathbf{r}}$ should also be an element living in the same relation-embedding space[1].
(2) **Closure**: to allow the possible existence of composition, in general, the elements $\mathbf{r}_1 \cdot \mathbf{r}_2$ should also be an element living in the same relation-embedding space[2].
(3) **Identity element**: the possibly existing inversion and composition together define another special and unique relation:

$$\mathbf{i} = \mathbf{r} \cdot \bar{\mathbf{r}}, \qquad \forall \mathbf{r} \in \mathcal{R}. \tag{6}$$

This element should map any entity to itself, and thus we call it *identity element*.
(4) **Associativity**: In a relational path with the length longer than three (containing three or more relations $\{r_1, r_2, r_3, ...\}$), as long as the sequential order does not change, the following two compositions should produce the same result:

$$(\mathbf{r}_1 \cdot \mathbf{r}_2) \cdot \mathbf{r}_3 = \mathbf{r}_1 \cdot (\mathbf{r}_2 \cdot \mathbf{r}_3). \tag{7}$$

The associativity is actually rooted in our definition of $\mathbf{r}_1 \cdot \mathbf{r}_2$ in (5) through the subsequent operating sequence in the entity space, from which, we can derive directly that:

$$\begin{aligned} \mathbf{O}_{(\mathbf{r}_1 \cdot \mathbf{r}_2) \cdot \mathbf{r}_3}\big[\,\cdot\,\big] &= \mathbf{O}_{\mathbf{r}_1 \cdot \mathbf{r}_2}\big[\mathbf{O}_{\mathbf{r}_3}[\cdot]\big] \\ &= \mathbf{O}_{\mathbf{r}_1}\big[\mathbf{O}_{\mathbf{r}_2}\big[\mathbf{O}_{\mathbf{r}_3}[\cdot]\big]\big] \\ &= \mathbf{O}_{\mathbf{r}_1}\big[\mathbf{O}_{\mathbf{r}_2 \cdot \mathbf{r}_3}[\cdot]\big] = \mathbf{O}_{\mathbf{r}_1 \cdot (\mathbf{r}_2 \cdot \mathbf{r}_3)}\big[\,\cdot\,\big], \end{aligned} \tag{8}$$

which then leads to the association (7). To help readers understand the practical meaning of associativity in real life cases, here we provide a simple example of the relational associativity:

$\mathbf{r}_1 =$ **isBrotherOf**, $\mathbf{r}_2 =$ **isMotherOf**, $\mathbf{r}_3 =$ **isFatherOf**.

Meanwhile, the following compositions are also meaningful:

$\mathbf{r}_1 \cdot \mathbf{r}_2 =$ **isUncleOf**, $\mathbf{r}_2 \cdot \mathbf{r}_3 =$ **isGrandmotherOf**.

In this example, one could easily see that:

$(\mathbf{r}_1 \cdot \mathbf{r}_2) \cdot \mathbf{r}_3 = \mathbf{r}_1 \cdot (\mathbf{r}_2 \cdot \mathbf{r}_3) =$ **isGranduncleOf**.

This is a simple demonstration of the associativity.
(5) **Commutativity/Nonconmmutativity**: In general, commuting two relations in a composition, i.e. $\mathbf{r}_1 \cdot \mathbf{r}_2 \leftrightarrow \mathbf{r}_2 \cdot \mathbf{r}_1$, may compose either the same or different results. We provide a simple illustrative examples for non-commutative compositions. Consider the following real world kinship:

$$\mathbf{r}_1 = \textit{isMotherOf}, \quad \mathbf{r}_2 = \textit{isFatherOf}. \tag{9}$$

Clearly, the composition $\mathbf{r}_1 \cdot \mathbf{r}_2$ and $\mathbf{r}_2 \cdot \mathbf{r}_1$ correspond to *isGrandmotherOf* and *isGrandfatherOf* relations respectively, which are different. This is a simple example of non- commutative cases. In real graphs, any cases may exist, and a proper embedding method should be able to accommodate both.

The first four properties are exactly the definition of a ***group***. In other words, *the group theory automatically emerges from the relational embedding problem itself, rather than being applied manually.* This is a quite convincing evidence that group theory is indeed the most natural language for relational embeddings if one aims at ultimately reproducing all possible local patterns in graphs. Besides, the fifth property on *commutativity/nonconmmutativity* are actually termed as **abelian/non-Abelian** in the group theory language. Since abelian is only a special case, to accommodate all possibilities, one should, in general, consider a **non-Abelian group for the relation embedding**, and guarantee at the same time it contains

[1] Given a graph, not all inversions correspond to meaningful relations, but an embedding model should be able to capture this possibility in general.
[2] Given a graph, not all compositions correspond to meaningful relations, but an embedding model should be able to capture this possibility in general.

at least one nontrivial abelian subgroup. We would term the corresponding embedding method as **NagE: the non-Abelian group embedding method**.

More explicitly, given a graph, to implement a group structure in embedding, one should embed all relations as group elements, which are parametrized by certain group parameters. For instance: the translation group T can be parametrized by a real number δ. And correspondingly, due to its vector nature, the embedding of entities could be regarded as a **representation (rep)** space of the same group. For the translation group, $\mathbb{R}$ (the real field) is a rep space of T.

This suggests the group representation theory is useful in knowledge graph embedding problems when talking about entity embeddings, and we leave this as a separate topic for subsequent works later. In the later section, we provide a general recipe for the graph embedding implementation.

3.3 Embedding models using different groups

In this section, we discuss embedding methods using different groups, from simple ones as T (the translation group) and $U(1)$, to complicated ones including $SU(2)$, $GL(n, \mathbb{V})$ (where $\mathbb{V}$ could be any type of fields), or even $Aff(\mathbb{V})$. It is important to note that, in practice, **continuous groups** are more reasonable than discrete ones, due to the two following reasons:

- The entity embedding space is usually continuous, which matches reps of the continuous group better. If used to accommodate a discrete group, a continuous space always contains infinite copies of irreducible reps of that group, which makes the analysis much more difficult.
- When training the embedding models, a gradient-based optimization search would be applied in the parameter space. However, different from continuous groups whose group parameter are also continuous, the parametrization of a discrete group uses discrete values, which brings in extra challenges for the training procedure.

With the two reasons above, we thus mainly consider continuous groups which are more reasonable choices. The other important feature of a group is commutativity, which we would mention for each group below. Besides the relational embedding group $\mathcal{G}$, the entity embedding space and the similarity measure also need to be determined. As discussed above, the entity embedding should be a proper rep space of $\mathcal{G}$. While for similarity measure $d(\cdot)$, we choose among the popular ones including L_p-norms (L_p) and the cos-similarity (cos), and a complete score function $s_{\mathbf{r}}(\mathbf{e_1}, \mathbf{e_2})$ for a triplet $(\mathbf{e_1}, \mathbf{r}, \mathbf{e_2})$ would be the distance from the relation-transformed head entity to the tail entity. One would notice many choices reproduce precedent works, and we show two examples below.

3.3.1 Example group: T. One could use n-copies of T, the translation group, for the relation embedding. This is a *noncompact abelian* group. The simplest rep-space would be the real field $\mathbb{R}$, which should also appear n times as $\mathbb{R}^n$. The group embedding then produces the following embedding vectors:

$$\begin{aligned} \mathbf{e} &\implies \vec{v}_{\mathbf{e}} = (x_1, x_2, \ \cdots, \ x_n), && \forall \mathbf{e} \in \mathcal{E}; \\ \mathbf{r} &\implies \vec{v}_{\mathbf{r}} = (\delta_1, \ \delta_2, \ \cdots, \ \delta_n), && \forall \mathbf{r} \in \mathcal{R}; \end{aligned}$$

both of which are n-dim. Here both x_i and δ_i are real numbers. In a triplet, the relation $\vec{v}_r$ acts as an addition vector added to the head entity $\mathbf{e_1}$. If one further chooses L_p-norm as the similarity measure, the complete score function $s_{\mathbf{r}}(\mathbf{e_1}, \mathbf{e_2})$ would be:

$$\|(\vec{v}_{\mathbf{r}} + \vec{v}_{\mathbf{e}_1}) - \vec{v}_{\mathbf{e}_2}\|_p, \tag{10}$$

this actually corresponds to the well-known **TransE** model [2]. There was a regularization in the original TransE model that changes the entity rep-space, which however has been removed in many later works by properly bounding the negative scores.

3.3.2 Example group: $U(1)$. One could use n-copies of $U(1)$, the 1-dim unitary transformation group, for the relational embedding. This is a *compact abelian* group. The simplest rep-space would be the real field $\mathbb{C}$, which should also appear n times as $\mathbb{C}^n$. The group embedding then produces the following embedding vectors:

$$\begin{aligned} \mathbf{e} &\implies \vec{v}_{\mathbf{e}} = (x_1, x_2, \ \cdots, \ x_n), && \forall \mathbf{e} \in \mathcal{E}; \\ \mathbf{r} &\implies \vec{v}_{\mathbf{r}} = (\phi_1, \phi_2, \ \cdots, \ \phi_n), && \forall \mathbf{r} \in \mathcal{R}; \end{aligned}$$

where x_i is a complex number containing a both real and imaginary part, while ϕ_i is a phase variable take values from 0 to 2π. Therefore the entity-embedding dimension is $2n$, while the relation dimension is n. In a triplet, the relation $\vec{v}_{\mathbf{r}}$ acts as a phase shift on the head entity $\mathbf{e}_1$. In a matrix form, one could define $\mathbf{R}_{\mathbf{r}}$ as the diagonal matrix with the i-th diagonal element being $e^{i\phi_i}$. If one further chooses L_p-norm as the similarity measure, the complete score function $s_{\mathbf{r}}(\mathbf{e_1}, \mathbf{e_2})$ would be:

$$\|\mathbf{R}_{\mathbf{r}} \cdot \vec{v}_{\mathbf{e}_1} - \vec{v}_{\mathbf{e}_2}\|_p = \|\left[e^{i\vec{v}_{\mathbf{r}}}\right] \circ \vec{v}_{\mathbf{e}_1} - \vec{v}_{\mathbf{e}_2}\|_p, \tag{11}$$

where $\circ$ means a Hadamard product. This precisely leads to the **RotatE** model [13].

On the other hand, one could also use the n-torus $\mathbb{T}^n$ as the rep-space:

$$\begin{aligned} \mathbf{e} &\implies \vec{v}_{\mathbf{e}} = (\theta_1, \theta_2, \ \cdots, \ \theta_n), && \forall \mathbf{e} \in \mathcal{E}; \\ \mathbf{r} &\implies \vec{v}_{\mathbf{r}} = (\phi_1, \phi_2, \ \cdots, \ \phi_n), && \forall \mathbf{r} \in \mathcal{R}; \end{aligned}$$

where θ_i represents a coordinate on the torus. Still using the L_p-norm similarity measure, the complete score function $s_{\mathbf{r}}(\mathbf{e_1}, \mathbf{e_2})$ now is:

$$\|\mathbf{R}_{\mathbf{r}} \cdot \vec{v}_{\mathbf{e}_1} - \vec{v}_{\mathbf{e}_2}\|_p = \|e^{i\vec{v}_{\mathbf{r}}} \circ e^{i\vec{v}_{\mathbf{e}_1}} - e^{i\vec{v}_{\mathbf{e}_2}}\|_p, \tag{12}$$

which leads to the **TorusE** model [5]. In the original implementation of TorusE, there is an additional projection π from $\mathbb{R}^n$ to $\mathbb{T}^n$.[3]

3.3.3 A summary of some example groups. We summarize the results of several chosen examples in Table 1.

Note in the Table 1, some groups have not been studied, but there are still some existing models which use a quite similar embedding space; while the major gap, between the existing models and their group embedding counterparts, is the constraint of group structures on the parametrization. For example, implementing group embedding with $GL(n, \mathbb{R})$, the n-dim general linear groups defined on field-$\mathbb{R}$, would lead to a model similar to RESCAL [12]. However, the original RESCAL model does not have a built-in group structure:

[3] Due to the special relation between T and $U(1)$, i.e. $U(1) \cong T/(2\pi\mathbb{Z})$, one could also regard TorusE as an implementation of group-embedding with T, which is more similar to the motivation in the original paper [5].

Group	Space	Abelian	$d(\cdot)$	Studied	Related Work
T	$\mathbb{R}^n$	YES	L_p	✓	TransE [2]
$U(1)$	$\mathbb{C}^n$	YES	L_p	✓	RotatE [13]
$U(1)$	$\mathbb{T}^n$	YES	L_p	✓	TorusE [5]
$SO(3)$	$\mathbb{R}^{3n}$	NO	L_p	–	–
$SU(2)$	$\mathbb{C}^{2n}$	NO	L_p	–	–
$GL(1,\mathbb{R})$	$\mathbb{R}^n$	YES	cos	✓	DisMult [17]
$GL(1,\mathbb{C})$	$\mathbb{C}^n$	YES	cos	✓	ComplEx [15]
$GL(n,\mathbb{R})$	$\mathbb{R}^n$	NO	cos	–	RESCAL [12]
$Aff(\mathbb{R}^n)$	$\mathbb{R}^n$	NO	L_p	–	TransR [9]
D_4	$\mathbb{R}^n$	NO	L_p	✓	DihEdral [16]

Table 1: Examples of the group embedding.

it uses arbitrary $n\times n$ real matrices, some of which may not be invertible, and hence are not group elements in $GL(n,\mathbb{R})$. It is, therefore, worth to add the extra invertible constraint in RESCAL, which requests matrices constructed through group parametrization rather than assigned arbitrary matrix elements. A similar analysis holds for the affine group $\mathrm{Aff}(\mathbb{R}^n)$.

4 GROUP EMBEDDING FOR KNOWLEDGE GRAPHS

In this section, we would firstly provide a general recipe for the group embedding implementation, and then provide two explicit examples of **NagE**, both of which apply a continuous non-Abelian group that has not been studied in any precedent works before.

4.1 A general group embedding recipe

We summarize the group embedding procedure as following:

(1) Given a graph, choose a proper group $\mathcal{G}$ for embedding. The choice may concern property of the task, such as commutativity and so on. And as stated above, in most general cases, a non-Abelian continuous group should be proper.
(2) Choose a rep-space for the entity embedding. For simplicity, one could use multiple (n) copies of the same rep ρ, which is the case of most existing works. Suppose ρ is a p-dim rep, then the total dimension of entity embedding would be pn, which is written as a vector $\vec{v}_{\mathbf{e}}$. Roughly speaking, k captures the relational structure and n encodes other feature.
(3) Choose a proper parametrization of $\mathcal{G}$, that is, choose a set of parameters indexing all group elements in $\mathcal{G}$. Suppose the number of parameters required to specify a group element is q, then the total dimension of relation embedding $\vec{v}_{\mathbf{r}}$ would be qn. A group element can now be expressed as a block-diagonal matrix $\mathbf{R}_{\mathbf{r}}$, with each block $\mathbf{M}_i$ being a $p\times p$ matrix whose entries are determined by the vector $\vec{v}_{\mathbf{r}}$.
(4) Choose a similarity measure $d(\cdot)$, the score value $s_{\mathbf{r}}(\mathbf{e}_1,\mathbf{e}_2)$ of a triplet $(\mathbf{e}_1,\mathbf{r},\mathbf{e}_2)$ is then:

$$s_{\mathbf{r}}(\mathbf{e}_1,\mathbf{e}_2) \equiv d\big(\mathbf{R}_{\mathbf{r}}\cdot\vec{v}_{\mathbf{e}_1},\ \ \vec{v}_{\mathbf{e}_2}\big) \tag{13}$$

Below we demonstrate the group embedding approach by implementing it with exampling continuous non-Abelian groups. As shown in table 1, two simple continuous non-Abelian groups that have not been studied are $SO(3)$ and $SU(2)$, we will implement them as relation embedding manifolds, which, as a result, produce two **NagE models**.

4.2 SO3E: NagE with group $SO(3)$

The 3D special orthogonal group $SO(3)$ is one of the simplest continuous non-Abelian group. As an illustrative demonstration, we construct an embedding model with $SO(3)$ structure and implement it in real experiments. Following the general recipe above, after determining the group $\mathcal{G}=SO(3)$, we choose a proper rep-space for entity embedding: $[\mathbb{R}^3]^{\otimes n}$, which consists n-copies of $\mathbb{R}^3$. Each $\mathbb{R}^3$ subspace transforms as the standard rep-space of $SO(3)$. All relations thus act as $3n\times 3n$ block diagonal matrix, with each block being a 3×3 complex matrix carrying the standard representation of $SO(3)$.

Next, we choose a proper parametrization of $SO(3)$. Instead of the more general angular momentum parametrization, due to our choice of using the standard representation, we could parameterize the $SO(3)$ elements using Euler angles (ϕ,θ,ψ), which is easier for implementation.

Put all together, our group embedding is then fixed as:

$$\begin{aligned}\mathbf{e} &\implies \vec{v}_{\mathbf{e}} = \big(x_1,y_1,z_1,\ \cdots,\ x_n,y_n,z_n\big), \quad \forall \mathbf{e}\in\mathcal{E};\\ \mathbf{r} &\implies \vec{v}_{\mathbf{r}} = \big(\phi_1,\theta_1,\psi_1,\ \cdots,\ \phi_n,\theta_n,\psi_n\big), \quad \forall \mathbf{r}\in\mathcal{R};\end{aligned}$$

both of which are $3n$-dim. In a triplet, the relation vector $\vec{v}_{\mathbf{r}}$ acts as a block diagonal matrix $\mathbf{R}_{\mathbf{r}}$, with each block matrix M_i acting in the subspace of (x_i,y_i,z_i):

$$\left[\begin{array}{c|c|c|c} \mathbf{M}_1 & 0 & \dots & 0 \\ \hline 0 & \mathbf{M}_2 & \dots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & \mathbf{M}_n \end{array}\right] \left[\begin{array}{c} x_1 \\ y_1 \\ z_1 \\ \hline x_2 \\ y_2 \\ z_2 \\ \hline \vdots \\ \hline x_n \\ y_n \\ z_n \end{array}\right]. \tag{14}$$

And each 3×3 block $\mathbf{M}_i$ is parametrized as following:

$$\begin{aligned}
\mathbf{M}_i^{11} &= \cos\psi_i \cos\phi_i - \cos\theta_i \sin\psi_i \sin\phi_i \\
\mathbf{M}_i^{12} &= \cos\psi_i \sin\phi_i + \cos\theta_i \cos\psi_i \cos\phi_i \\
\mathbf{M}_i^{13} &= \sin\psi_i \sin\theta_i \\
\mathbf{M}_i^{21} &= -\sin\psi_i \cos\phi_i - \cos\theta_i \sin\psi_i \cos\phi_i \\
\mathbf{M}_i^{22} &= -\sin\psi_i \sin\phi_i + \cos\theta_i \cos\psi_i \cos\phi_i \\
\mathbf{M}_i^{23} &= \cos\psi_i \sin\theta_i \\
\mathbf{M}_i^{31} &= \cos\psi_i \sin\theta_i \\
\mathbf{M}_i^{32} &= -\cos\psi_i \cos\theta_i \\
\mathbf{M}_i^{33} &= \cos\theta_i
\end{aligned} \tag{15}$$

4.3 SU2E: NagE with group $SU(2)$

The 2D special unitary group $SU(2)$ is another simple continuous non-Abelian group. We choose the rep-space for entity embedding as: $[\mathbb{C}^2]^{\otimes n}$, which consists n-copies of $\mathbb{C}^2$. Each $\mathbb{C}^2$ subspace transforms as the standard rep-space of $SU(2)$. All relations thus act as $2n \times 2n$ block diagonal matrix, with each block being a 2×2 complex matrix carrying the standard representation of $SU(2)$.

Next, we choose a proper parametrization of $SU(2)$. An analysis with the corresponding Lie algebra $\mathfrak{su}(2)$ shows that any group element could be written as [6]:

$$e^{i\alpha[\hat{\mathbf{n}}\cdot\vec{\mathbf{J}}]} = \cos\alpha\hat{1} + i\sin\alpha\hat{\mathbf{n}}\cdot\vec{\mathbf{J}}, \tag{16}$$

where α is a rotation angle taken from $[0, 2\pi]$, and $\hat{\mathbf{n}}$ is a unit vector on S^2, represented by two other angles (θ, ϕ); moreover, the symbol $\hat{1}$ means an identity matrix, and $\vec{\mathbf{J}}$ are three generators of the group: $(\mathbf{J}_x, \mathbf{J}_y, \mathbf{J}_z)$, which, in the standard rep have the following form:

$$J_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ J_y = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix},\ J_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Put all together, our group embedding is then fixed as:

$$\begin{aligned}
\mathbf{e} &\implies \vec{v}_{\mathbf{e}} = (x_1, y_1,\ x_2, y_2,\ \cdots,\ x_n, y_n), \quad \forall \mathbf{e} \in \mathcal{E}; \\
\mathbf{r} &\implies \vec{v}_{\mathbf{r}} = (\alpha_1, \hat{\mathbf{n}}_1, \alpha_2, \hat{\mathbf{n}}_2,\ \cdots, \alpha_n, \hat{\mathbf{n}}_n), \quad \forall \mathbf{r} \in \mathcal{R};
\end{aligned}$$

where x_i and y_i are complex numbers, and α_i and $\hat{\mathbf{n}}_i = (\theta_i, \phi_i)$ represent angles. In a triplet, the relation $\vec{v}_{\mathbf{r}}$ acts as a block diagonal matrix $\mathbf{R}_{\mathbf{r}}$, with each block matrix M_i acting in the subspace of (x_i, y_i). And each 2×2 block $\mathbf{M}_i$ is parametrized as [6]:

$$\begin{bmatrix} \cos\alpha_i + i\sin\alpha_i \sin\theta_i & ie^{-i\phi_i} \cdot \sin\alpha_i \cos\theta_i \\ ie^{-i\phi_i} \cdot \sin\alpha_i \cos\theta_i & \cos\alpha_i - i\sin\alpha_i \sin\theta_i \end{bmatrix}.$$

4.4 Similarity measure and loss function

We choose L_2-norm as the similarity measure $d(\cdot)$ to compute the score value:

$$s_{\mathbf{r}}(\mathbf{e}_1, \mathbf{e}_2) = \|\mathbf{R}_{\mathbf{r}} \cdot \vec{v}_{\mathbf{e}_1} - \vec{v}_{\mathbf{e}_2}\|_2 \tag{17}$$

We design the model loss function for a triple $(\mathbf{e_1}, \mathbf{r}, \mathbf{e_2})$ as follows:

$$\begin{aligned}
L = &-\log\sigma\left[\gamma - s_{\mathbf{r}}(\mathbf{e_1}, \mathbf{e_2})\right] \\
&- \sum_{i=1}^{n} p\left(\mathbf{e'_{1i}}, \mathbf{r}, \mathbf{e'_{2i}}\right) \log\sigma\left[s_r\left(\mathbf{e_{1i}}', \mathbf{e_{2i}}'\right) - \gamma\right] \\
p\left(\mathbf{e'_{1j}}, \mathbf{r}, \mathbf{e'_{2j}} \middle| \{(\mathbf{e_{1i}}, \mathbf{r}, \mathbf{e_{2i}})\}\right) &= \frac{e^{\alpha[\gamma - s_{\mathbf{r}}(\mathbf{e'_{1j}}, \mathbf{e'_{2j}})]}}{\sum_i e^{\alpha[\gamma - s_{\mathbf{r}}(\mathbf{e'_{1i}}, \mathbf{e'_{2i}})]}}
\end{aligned}$$

where σ is the Sigmoid function, γ is the margin used to prevent over-fitting. $\mathbf{e'_{1i}}$ and $\mathbf{e'_{2i}}$ are negative samples while $p\left(\mathbf{e'_{1i}}, r, \mathbf{e'_{2i}}\right)$ is the adversarial sampling mechanism with temperature α we adopt self-adversarial negative sampling setting from [13]. We term the resulting model as **SO3E** and **SU2E** respectively for the above two groups. We mention other implementation details in the next section.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets: The most popular public knowledge graph datasets include FB15K [1] and WN18 [10]. FB15K-237 [14] and WN18RR [4] datasets were derived from these two, in which the inverse relations were removed. FB15K dataset is a huge knowledge base with general facts containing 1.2 billion instances of more than 80 million entities. For benchmarking, usually, a frequency filter was applied to obtain occurrence larger than 100 resulting in 592,213 instances with 14,951 entities and 1,345 relation types. WN18 was extracted from WordNet [10] dictionary and thesaurus, the entities are word senses and the relations are lexical relations between them. It has 151,442 instances with 40,943 entities and 18 relation types.

Evaluation Protocols: We use three categories of protocols for evaluations, namely, cut-off Hit ratio (H@N), Mean Rank(MR) and Mean Reciprocal Rank (MRR). H@N measures the ratio of correct entities predictions at a top n prediction result cut-off. Following the baselines used in recent literature, we chose $n = 1, 3, 10$. MR evaluates the average rank among all the correct entities. MRR is the average rank inverse rank of the correct entities.

Implementation Details: We implemented our models using pytorch[4] framework and experimented on a server with an Nvidia Titan-1080 GPU. The Adam [8] optimizer was used with the default β_1 and β_2 settings. A learning rate scheduler observing validation loss decrease was used to reduce learning rate by half after patience of 3000. Batch-size was set at 1024. We did a grid search on the following hyper-parameters: embedding dimension $d \in \{100, 250, 400, 500\}$; learning rate $\eta \in \{3e-4, 1e-4, 3e-5, 1e-5, 3e-6\}$; number of negative samples during training $n_{neg} \in \{128, 256, 512\}$; adversarial negative sampling temperature $\alpha \in \{0.5, 0.75, 1.0, 1.25\}$; loss function margin $\gamma \in \{6, 9, 12, 20, 22, 24, 26\}$.

5.2 Results and Model analysis

Empirical results on FB15k and WN18 are reported in Table 2. We compared the embedding results of different groups, including T, $U(1)$, $GL(1, \mathbb{R})$, $GL(1, \mathbb{C})$, $SO(3)$ and $SU(2)$, which are mainly categorized by the commutativity. As discussed in Sec. 3.3, the

[4] https://www.pytorch.org

Group	Commutativity	WN18				FB15k				Example
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	
T	Abelian	0.495	0.113	0.888	0.943	0.463	0.297	0.578	0.749	TransE
U(1)	Abelian	0.949	**0.944**	0.952	0.959	**0.797**	**0.746**	0.830	0.884	RotatE
U(1)	Abelian	0.947	0.943	0.950	0.954	0.733	0.674	0.771	0.832	TorusE
GL(1, $\mathbb{R}$)	Abelian	0.822	0.728	0.914	0.936	0.654	0.546	0.733	0.824	DistMult
GL(1, $\mathbb{C}$)	Abelian	0.946	0.942	0.949	0.954	0.692	0.599	0.759	0.840	ComplEx
SO(3)	non-Abelian	**0.950**	**0.944**	0.953	**0.960**	0.794	0.740	**0.831**	**0.886**	SO3E
SU(2)	non-Abelian	**0.950**	**0.944**	**0.954**	**0.960**	0.791	0.734	**0.831**	**0.886**	SU2E

Table 2: Link prediction on WN18 and FB15k (bold represent the best scores, underlined represent the second best).

Group	Commutativity	WN18RR				FB15k-237				Example
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	
T	Abelian	0.226	-	-	0.501	0.294	-	-	0.465	TransE
U(1)	Abelian	0.476	0.428	0.492	0.571	0.338	0.241	0.375	**0.533**	RotatE
GL(1, $\mathbb{R}$)	Abelian	0.430	0.390	0.440	0.490	0.241	0.155	0.263	0.419	DistMult
GL(1, $\mathbb{C}$)	Abelian	0.440	0.410	0.460	0.510	0.247	0.158	0.275	0.428	ComplEx
SO(3)	non-Abelian	**0.477**	**0.432**	**0.493**	0.574	**0.340**	**0.244**	**0.378**	0.530	SO3E
SU(2)	non-Abelian	0.476	0.429	**0.493**	**0.575**	**0.340**	0.243	0.376	0.532	SU2E

Table 3: Link prediction on WN18RR and FB15k-237 (bold represent the best scores, underlined represent the second best).

former four groups have been implicitly applied in existing models. For $SO(3)$ and $SU(2)$, we report the result of our own experiments. Results of the other models are taken from their original literature: TransE using group T was proposed in [2]; RotatE using group $U(1)$ was proposed in [13] while TorusE with the same group was proposed in [5]; group $GL(1, \mathbb{V})$ was implemented in DisMult [17] with $\mathbb{V} = \mathbb{R}$ and in ComplEx [15] with $\mathbb{V} = \mathbb{C}$.

Results on datasets FB15K-237 and WN18RR are demonstrated in Table 3 respectively. We remove TorusE from the tables due to the absence of results in the original work, and refer to [11] for TransE.

In the FB15k dataset, the main hyper-relation is anti-/symmetry and inversion. The dataset has a vast amount of unique entities. Shown in Table 2, the RotatE model achieved good performance in this dataset. SO3E and SU2E achieved comparable result across the metrics. On the other hand, since inversion relations are removed in FB15k-237, the dominant portion of hyper-relations becomes the composition. We can see RotatE fail on this task due to non-Abelian hyper-relations. Shown in Table 3, the continuous non-Abelian group method SO3E and SU2E outperformed most of the metrics.

In the WN18 dataset, SO3E and SU2E outperformed all the baselines on all metrics shown in Table 2. The WN18RR dataset removes the inversion relations from WN18, left only 11 relations and most of them are symmetry patterns. We can see from Table 3, SO3E and SU2E model performed well due to their non-Abelian nature.

Drawn from the experiments, two factors significantly impact the embedding model performance: the embedding dimension, and group attributes (including commutativity and continuity). As theoretically analyzed in Section 3.2, and empirically shown above, continuous non-Abelian groups are more reasonable choices for general tasks. It is important to note that SO3E and SU2E proposed above are exampling models for our group embedding framework, and they use the simplest continuous non-Abelian groups. Much more efforts could be devoted in this direction in the future.

6 CONCLUSION AND FUTURE WORK

We proved for the first time the emergence of a group definition in the KG representation learning. This proof suggests that relational embeddings should respect the group structure. A novel theoretic framework based on group theory was therefore proposed, termed as the *group embedding* of relational KGs. Embedding models designed based on our proposed framework would automatically accommodate all possible hyper-relations, which are building-blocks of the link prediction task.

From the group-theoretic perspective, we categorize different embedding groups regarding commutativity and the continuity and empirically compared their performance. We also realize that many recent models correspond to embeddings using different groups. Generally speaking, a continuous non-Abelian group embedding should be powerful for a generic KG completion task. We demonstrate this idea by examining two simple exampling models: **SO3E** and **SU2E**. With $SO(3)$ and $SU(2)$ as embedding groups, our models showed promising performance in challenging tasks where hyper-relations become crucial.

In the proposed framework, beside embedding relations as group elements, entity embeddings live in different representation space of the corresponding group. And therefore an investigation of group representation theory in entity embedding is highly demanded. We leave this in future works. On the other hand, although empirical evaluations focus on linear models, it is important to note that the proof of the group structure only relies on the KG task itself. This means our conclusion also works for more general models, including neural-network-based ones. Beyond KG embeddings, the same analysis could be applied to other representation learning where

intrinsic relational structures are prominent. An implementation of group structures in more general cases would be very interesting.

ACKNOWLEDGEMENT

Funding for the shared GPU-computing facility used in this research was provided by NSF OAC 1920147.

REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *In SIGMOD Conference*. 1247–1250.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13)*. Curran Associates Inc., USA, 2787–2795. http://dl.acm.org/citation.cfm?id=2999792.2999923

[3] Chen Cai. 2019. Group Representation Theory for Knowledge Graph Embedding. *arXiv preprint arXiv:1909.05100* (2019).

[4] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*. 1811–1818. https://arxiv.org/abs/1707.01476

[5] Takuma Ebisu and Ryutaro Ichise. 2018. TorusE: Knowledge Graph Embedding on a Lie Group. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 1819–1826. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16227

[6] B. Hall and B.C. Hall. 2003. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer. https://books.google.com/books?id=m1VQi8HmEwcC

[7] Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, and Jun Zhao. 2017. An End-to-End Model for Question Answering over Knowledge Base with Cross-Attention Combining Global Knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 221–231. https://doi.org/10.18653/v1/P17-1021

[8] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2014).

[9] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 2181–2187. http://dl.acm.org/citation.cfm?id=2886521.2886624

[10] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (Nov. 1995), 39–41. https://doi.org/10.1145/219717.219748

[11] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2017. A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121* (2017).

[12] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-way Model for Collective Learning on Multi-relational Data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11)*. Omnipress, USA, 809–816. http://dl.acm.org/citation.cfm?id=3104482.3104584

[13] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HkgEQnRqYQ

[14] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Beijing, China, 57–66. https://doi.org/10.18653/v1/W15-4007

[15] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, 2071–2080. http://dl.acm.org/citation.cfm?id=3045390.3045609

[16] Canran Xu and Ruijiang Li. 2019. Relation Embedding with Dihedral Group in Knowledge Graph. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 263–272. https://doi.org/10.18653/v1/P19-1026

[17] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *CoRR* abs/1412.6575 (2014).

[18] Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*. 2731–2741.