Dual Complex Number Knowledge Graph Embeddings

Yao Dong[1], **Qingchao Kong(✉)**[1,2,3], **Lei Wang**[1], **Yin Luo**[1]
Beijing Wenge Technology Co., Ltd.[1]
State Key Laboratory of MAIS, Institute of Automation, Chinese Academy of Sciences[2]
School of Artificial Intelligence, University of Chinese Academy of Sciences[3]
{yao.dong, lei.wang, yin.luo}@wenge.com, qingchao.kong@ia.ac.cn

Abstract

Knowledge graph embedding, which aims to learn representations of entities and relations in large scale knowledge graphs, plays a crucial part in various downstream applications. The performance of knowledge graph embedding models mainly depends on the ability of modeling relation patterns, such as symmetry/antisymmetry, inversion and composition (commutative composition and non-commutative composition). Most existing methods fail in modeling the non-commutative composition patterns. Several methods support this kind of pattern by modeling in quaternion space or dihedral group. However, extending to such sophisticated spaces leads to a substantial increase in the amount of parameters, which greatly reduces the parameter efficiency. In this paper, we propose a new knowledge graph embedding method called dual complex number knowledge graph embeddings (DCNE), which maps entities to the dual complex number space, and represents relations as rotations in 2D space via dual complex number multiplication. The non-commutativity of the dual complex number multiplication empowers DCNE to model the non-commutative composition patterns. In the meantime, modeling relations as rotations in 2D space can effectively improve the parameter efficiency. Extensive experiments on multiple benchmark knowledge graphs empirically show that DCNE achieves significant performance in link prediction and path query answering.

Keywords: knowledge graph embedding, dual complex number, 2D rotation

1. Introduction

Knowledge graphs (KGs) contain structured facts of the real world. Large-scale KGs such as WordNet (Miller, 1995), YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008) and Nell (Mitchell et al., 2018) have been applied to several downstream tasks including machine translation (Zhao et al., 2020), relation extraction (Vashishth et al., 2018), question answering (Hao et al., 2017), conversation generation (Zhou et al., 2018) and recommender systems (Zhang et al., 2016). KGs may include millions or even billions of triplets. However, real-world KGs are usually incomplete, i.e., having numerous missing links (Socher et al., 2013). Therefore, predicting missing links by mining information in KG has gained growing interest in recent years. This task can be further divided into two categories according to the length of the predicting path: link prediction ($length = 1$) and path query answering ($length \geq 1$). Link prediction focuses on single-hop reasoning (e.g., answering the query $\mathrm{s} \rightarrow \mathrm{r} \rightarrow ?$, where r is a relation), while path query answering (PQA) is interested in multi-hop reasoning (e.g., answering the path query $\mathrm{s} \rightarrow \mathrm{path} \rightarrow ?$, where path contains multiple relations).

A promising approach for link prediction and PQA is knowledge graph embedding (KGE), which encodes each element in KG into a continuous low-dimensional vector space, and predicts missing links by evaluating the scores of them. The performance of KGE methods greatly relys on the ability of modeling and inferring relation patterns including symmetry/antisymmetry, inversion and composition. Actually, the composition patterns can be further divided into the **commutative composition patterns** and **non-commutative composition patterns**. For example, "spouse" is a symmetric relation and "father" is an antisymmetric relation. Relations such as "has_part" and "part_of" forms an inversion pattern. The meaning of the composition of "father" and "spouse" depends on the relative order, which is a non-commutative composition pattern. (An example is shown in Fig. 1.) By contrast, the composition of "father" and "father" has a definite meaning, i.e.,"grandfather", which is a commutative composition pattern.

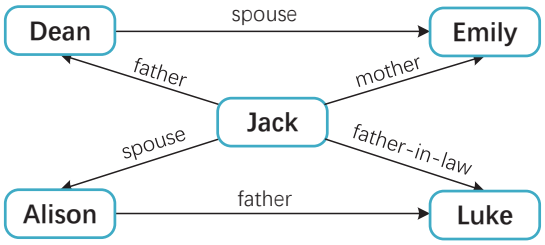


Figure 1: An example in real world. Jack's father's spouse is Emily, i.e., Jack's mother; Jack's spouse's father is Luke, i.e., Jack's father-in-law. "father" and "spouse" form a non-commutative composition pattern.

Current methods can capture one or more re-

LREC-COLING 2024, pages 5391–5400

lation patterns. For instance, TransE (Bordes et al., 2013) models antisymmetry, inversion and commutative composition patterns by translating relations from head entities to tail entities. RotatE (Sun et al., 2019), which regards relations as rotations in complex space, aims to model the symmetry/antisymmetry, inversion and commutative composition patterns. However, most approaches fail to model the non-commutative composition patterns, which are essential for learning semantics.

By the non-commutativity of quaternion and dual quaternion, methods such as QuatE (Zhang et al., 2019), Rotate3D (Gao et al., 2020) and DualE (Cao et al., 2021) can model the non-commutative composition patterns. Nevertheless, quaternion-valued methods model relations as **rotations in 3D space**, which sharply **increases the space cost** than that in 2D space.

In this paper, we propose a new method called **DCNE** for knowledge graph embedding. DCNE projects entities to the dual complex number space, and represents relations as rotations around any point in 2-dimensional Euclidean space via dual complex number multiplication. The non-commutativity of the dual complex number multiplication empowers DCNE to model the non-commutative composition patterns. In addition, modeling relations as rotations in 2D space can effectively improve the parameter efficiency.

In summary, our contributions are listed as follows: (1) DCNE provides a balanced solution to model the non-commutative composition patterns, which reduces the space cost than existing KGE methods. (2) To the best of our knowledge, this paper is the first to introduce dual complex numbers into knowledge graph embeddings, which shows a new perspective on modeling the non-commutative composition patterns. (3) We provide comprehensive theoretical analyses of DCNE including inference patterns and parameter efficiency. (4) Experimental results verify the superiority of DCNE in link prediction and path query answering.

The code and supplemental materials are publicly available at GitHub[1].

2. Related Work

In this section, we roughly divide the existing KGE methods into three categories based on the non-commutativity and discuss their connections to our approach.

Models without Non-commutativity. Most existing KGE methods lack the non-commutativity, which is the key to modeling non-commutative composition patterns. TransE (Bordes et al., 2013) is the most representative KGE model, which encodes both entities and relations as vectors in embedding space based on the principle $\mathrm{h} + \mathrm{r} \approx \mathrm{t}$, where h, r, t denote the head entity, relation and tail entity, respectively. To remedy the limitations of TransE, several variants (Ji et al., 2015; Lin et al., 2015; Wang et al., 2014; Xiao et al., 2016) are proposed to improve the performance when modeling 1-N, N-1 and N-N relations. Beyond Euclidean space, TorusE (Ebisu and Ichise, 2018) models triplets on a torus. Inspired by Euler's identity $e^{i\theta} = cos\theta + isin\theta$, RotatE (Sun et al., 2019) regards translations as rotations from head entities to tail entities in complex space. RotH (Chami et al., 2020) introduces a promising method which models elements in hyperbolic space. Rot-Pro (Song et al., 2021) supports the transitivity patterns by combining the projection and relational rotation. In addition, BoxE (Abboud et al., 2020) encodes elements by explicitly defining the region as boxes. RESCAL (Nickel et al., 2011) is the first bilinear model that can perform collective learning via matching the latent semantics between entities and relations. DistMult (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) are proposed to solve the overfitting problem of RESCAL. HolE (Nickel et al., 2016) absorbs the quintessence from DistMult and ComplEx. Approaches such as SimplE (Kazemi and Poole, 2018) and TuckER (Balazevic et al., 2019b) turn to different forms of decompositions. Recently, to increase the modeling capacity, HousE (Li et al., 2022) utilizes Householder transformations. Although some of these methods claim the ability to model the composition patterns, in practice, they model the commutative composition patterns, not non-commutative composition patterns.

Models with Non-commutativity. Several approaches extend the vector space to more sophisticated spaces to obtain the non-commutativity, which is important for multi-hop reasoning. Specifically, Rotate3D (Gao et al., 2020) and QuatE (Zhang et al., 2019) model relations as rotations in quaternion space with different score functions. DualE (Cao et al., 2021) makes the first attempt to combine rotation and translation by extending the embedding space to dual quaternion space. RotateCT (Dong et al., 2022) provides a novel use of translation in which the coordinate transformation and translation are integrated in complex space. However, the performance of RotateCT on certain datasets stems from the self-adversarial negative sampling (Sun et al., 2019). In addition, DihEdral (Xu and Li, 2019) limits relation matrices to be block diagonal and represents each block with an element in a dihedral group. Although such approaches can model the non-commutative composition patterns, the space cost of them are greatly increased due to the complicated vector

[1] https://github.com/JensenDong/DCNE

spaces such as quaternion space. Our method DCNE models relations as rotations around any point in 2D space via dual complex numbers. Compared to quaternion-valued methods, DCNE can significantly improve the parameter efficiency.

Indeterminate Models. A number of researchers focus on utilizing the neural networks. However, the neural networks lack interpretability, and it is difficult to give theoretical analyses from the perspective of the non-commutativity. Generally, such approaches verify the performance by empirical experiments. ConvE (Dettmers et al., 2018), ConvKB (Nguyen et al., 2018) and InteractE (Vashishth et al., 2020) capture the interactions between entities and relations by convolutional neural networks. R-GCN (Schlichtkrull et al., 2018) and KBGAT (Nathani et al., 2019) redesign a graph convolutional network and a graph attention network, respectively. Recent works try to exploit more global graph structures like multi-hop paths. Path-RNN (Das et al., 2017) and ROP (Yin et al., 2018) employ RNNs to explicitly model paths. CoKE (Wang et al., 2019) uses a stack of Transformer blocks to model paths. Compared to the above models, our method DCNE is more interpretable with comprehensive theoretical analyses.

3. Preliminaries

In this section, we introduce dual complex numbers, along with its operations and properties. Then, we show how to use dual complex numbers to represent rotations in 2D space.

3.1. Dual Complex Numbers

A dual complex number q is of the form: $q = A + Bi + C\epsilon + Di\epsilon$, where A, B, C and D are real numbers. The set $\{1, i, \epsilon, i\epsilon\}$ forms a basis of the vector space of dual complex numbers. Imaginary unit i anti-commutes with dual unit ϵ. i and ϵ satisfy the following rules: $i^2 = -1, \epsilon^2 = 0, i\epsilon i = \epsilon, \epsilon i = -i\epsilon$. Thus, the multiplication of two dual complex numbers q_1 and q_2 is defined as:

$$\begin{aligned} q_1 \otimes q_2 = &(A_1A_2 - B_1B_2) + (A_1B_2 + B_1A_2)i \\ &+ (C_1A_2 + D_1B_2 + A_1C_2 - B_1D_2)\epsilon \\ &+ (D_1A_2 - C_1B_2 + B_1C_2 + A_1D_2)i\epsilon \end{aligned} \quad (1)$$

Apparently, the multiplication between dual complex numbers is non-commutative. If and only if $\frac{B_1}{B_2} = \frac{C_1}{C_2} = \frac{D_1}{D_2}, q_1 \otimes q_2 = q_2 \otimes q_1$.

Conjugate The conjugate of a dual complex number q is defined as $\bar{q} = A - Bi - C\epsilon - Di\epsilon$.

Norm The norm of a dual complex number q is defined as $\|q\| = \sqrt{A^2 + B^2}$.

Inversion The inversion of a dual complex number q is defined as $q^{-1} = \frac{\bar{q}}{\|q\|^2}$.

3.2. Representing Rotations in 2D Space via Dual Complex Numbers

The Euclidean plane can be represented by the set $\Pi = \{i + x\epsilon + yi\epsilon | x \in \mathbb{R}, y \in \mathbb{R}\}$. An element $v = i + a\epsilon + bi\epsilon$ on Π represents the point on the Euclidean plane with cartesian coordinate (a, b).

Theorem 1. *If $q = cos\frac{\theta}{2} + sin\frac{\theta}{2}(i + x\epsilon + yi\epsilon)$, $v \in \Pi$, then $v' = qvq^{-1}$ is the result of v rotating θ around the point (x, y) in Euclidean plane.*

4. Methodology

4.1. Dual Complex Number Representations for Knowledge Graph Embeddings

Formally, let $\mathcal{E}$ denote the set of entities and $\mathcal{R}$ denote the set of relations. Then a knowledge graph $\mathcal{G}$ is a collection of factual triplets $\{(\mathrm{h}, \mathrm{r}, \mathrm{t})\}$, where $\mathrm{h}, \mathrm{t} \in \mathcal{E}$ and $\mathrm{r} \in \mathcal{R}$. Lowercase letters h, r and t denote the head entity, relation and tail entity, respectively; the corresponding bold letters $\mathbf{h}, \mathbf{r}$ and $\mathbf{t}$ denote the embeddings of them. Note that the i-th element of $\mathbf{h}$ is h_i. Let k denote the dimension of entity and relation embeddings.

In this paper, we propose DCNE to model the non-commutative composition patterns in 2D space. Our model projects entities to the dual complex number space, i.e., $\mathbf{h}, \mathbf{t} \in \mathbb{DC}^k$. Then each element of an entity $\mathbf{e}$ is of the form $e_i = i + x_i\epsilon + y_i i\epsilon$, which represents a point (x_i, y_i) in 2D space. Further, we define each relation as an element-wise rotation around $\{(x_i, y_i)\}$ from head entity $\mathbf{h}$ to tail entity $\mathbf{t}$. In other words, given a golden triplet $(\mathrm{h}, \mathrm{r}, \mathrm{t})$, we expect that:

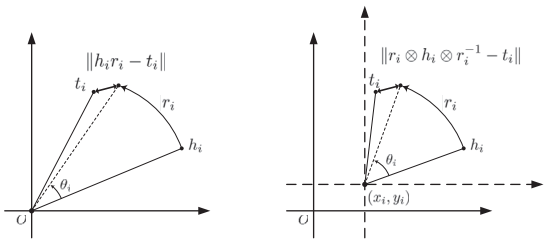
$$\mathbf{t} = \mathbf{r} \circ \mathbf{h} \circ \mathbf{r}^{-1} \quad (2)$$

where $\circ$ is the Hadmard (or element-wise) product. Specifically, we have $t_i = r_i \otimes h_i \otimes r_i^{-1}$ for each element of $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$. Here, we constrain the modulus of each element of $\mathbf{r} \in \mathbb{DC}^k$, i.e., $r_i \in \mathbb{DC}$, to be $|r_i| = 1$. Then r_i is of the form $r_i = cos\frac{\theta_{r,i}}{2} + sin\frac{\theta_{r,i}}{2}(i + x_{r,i}\epsilon + y_{r,i}i\epsilon)$, which corresponds a counterclockwise rotation by $\theta_{r,i}$ radians around the point $(x_{r,i}, y_{r,i})$. We define the distance-based score function as follows:

$$d_r(\mathrm{h}, \mathrm{t}) = \sum_{i=1}^{k} \|r_i \otimes h_i \otimes r_i^{-1} - t_i\| \quad (3)$$

Optimization. We use a loss function similar to the negative sampling loss (Mikolov et al., 2013) for optimizing:

$$\begin{aligned} L = &-\sum_{i=1}^{m} \frac{1}{m} \log \sigma(d_r(\mathrm{h}'_i, \mathrm{t}'_i) - \gamma) - \log \sigma(\gamma - d_r(\mathrm{h}, \mathrm{t})) \\ &+ \lambda(\|\mathbf{h}\|^2 + \|\mathbf{t}\|^2) \end{aligned} \quad (4)$$



(a) RotatE (b) DCNE

Figure 2: Illustrations of RotatE (a) and DCNE (b). RotatE models r as a rotation around the origin in complex plane. DCNE models r as a rotation around any point in Euclidean plane.

where γ is a fixed margin, σ is the sigmoid function, m is the negative sampling size, $d_r(\mathrm{h}'_i, \mathrm{t}'_i)$ represents the score of i-th negative triplet and λ is the regularization rate of **h** and **t**. We utilize Adam (Kingma and Ba, 2014) as the optimizer and use the self-adversarial negative sampling (Sun et al., 2019) for generating negative samples.

4.2. Discussion

In this part, we introduce the inference patterns and provide some theoretical analyses of DCNE. Then, we discuss the connections between DCNE and RotatE, and give a comparison about the number of free parameters among DCNE, Rotate3D and QuatE.

Inference Patterns. Knowledge graphs mainly consist of three important relation patterns: symmetry/antisymmetry, inversion and composition (commutative composition and non-commutative composition).

Theorem 2. *(**Inference ability**) DCNE can infer the **symmetry/antisymmetry**, **inversion** and **composition** patterns.*

Based on Theorem 2, we have two corollaries for symmetry/antisymmetry and inversion patterns.

Corollary 1. *If r is a symmetric relation, then $\theta_i = 0 \vee \pm\pi$; if r is an antisymmetric relation, then $\theta_i \neq 0, \pm\pi$.*

Corollary 2. *If r_1 is the inverse of r_2, then $\theta_{1,i} + \theta_{2,i} = 0 \vee \pm 2\pi$.*

DCNE v.s. RotatE. DCNE encodes relations as rotations around any point in 2-dimensional Euclidean space, whereas RotatE represents relations as rotations around the origin in complex plane. Complex plane is a kind of 2D space. Essentially, both RotatE and DCNE regard relations as rotations in 2D space. Hence, RotatE can be viewed as a special case of DCNE when the center of rotation is the origin. Fig. 2 provides illustrations of RotatE and DCNE.

Parameter Efficiency. The number of free parameters on multiple datasets are shown in Table 2, from which we can see that DCNE reduces up to 30% parameters against Rotate3D on all datasets. Intuitively, Rotate3D models relations in 3D space, whereas DCNE model relations as rotations in 2D space via dual complex numbers. Representing a point in 2D space is one parameter less than that in 3D space, which reduces the space cost of DCNE.

5. Experiments

In this section, we evaluate DCNE on link prediction (Bordes et al., 2013) and path query answering (Guu et al., 2015).

5.1. Link Prediction

Link prediction aims to predict the missing h or t for a triplet $(\mathrm{h}, \mathrm{r}, \mathrm{t})$, namely, predicting the head query $? \rightarrow \mathrm{r} \rightarrow \mathrm{t}$ or the tail query $\mathrm{h} \rightarrow \mathrm{r} \rightarrow ?$.

Datasets. We evaluate DCNE on four widely used benchmarks: WN18 (Bordes et al., 2013), FB15k (Bordes et al., 2013), WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova and Chen, 2015). WN18 is a subset of WordNet, consisting of lexical relations between words. FB15k is extracted from Freebase, a large-scale knowledge graph containing general facts. The main relation patterns in WN18 and FB15k are **symmetry/antisymmetry** and **inversion** (Sun et al., 2019). However, both WN18 and FB15k suffer from test leakage through inverse relations (Toutanova and Chen, 2015). To avoid this problem, WN18RR and FB15k-237 remove the inverse relations in WN18 and FB15k. Thus, the main relation patterns in WN18RR and FB15k-237 are **symmetry/antisymmetry** and **composition** (Sun et al., 2019).

Evaluation Protocol. We report three standard evaluation metrics: Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hits@N, where $N = 1, 3, 10$. MR is the average rank of all correct entites. MRR is the mean reciprocal rank of all correct entities. Hits@N represents the proportion of correct entities whose rank is not larger than N. Note that an excellent KGE model should achieve a lower MR, a higher MRR and a higher Hits@N. Following Bordes et al. (2013), we report the filtered results to avoid possibly flawed evaluation.

Implementation Details. We use PyTorch to implement our model and test it on a single GPU (Tesla V100).

Main Results. We compare DCNE with a number of strong baselines. For models without non-commutativity, we report TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), SimplE (Kazemi and Poole, 2018),

	WN18					FB15k				
Model	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TransE (2013)	-	0.495	0.113	0.888	0.943	-	0.463	0.297	0.578	0.749
DistMult (2015)	655	0.797	-	-	0.946	42	**0.798**	-	-	**0.893**
ComplEx (2016)	-	0.941	0.936	0.945	0.947	-	0.692	0.599	0.759	0.840
SimplE (2018)	-	0.942	0.939	0.944	0.947	-	0.727	0.660	0.773	0.838
TorusE (2018)	-	0.947	0.943	0.950	0.954	-	0.733	0.674	0.771	0.832
RotatE (2019)	309	0.949	0.944	0.952	0.959	40	0.797	**0.746**	0.830	0.884
ConvE (2018)	374	0.943	0.935	0.946	0.956	51	0.657	0.558	0.723	0.831
R-GCN+ (2018)	-	0.819	0.697	0.929	**0.964**	-	0.696	0.601	0.760	0.842
NKGE (2018)	336	0.947	0.942	-	0.957	56	0.730	0.650	0.790	0.871
DihEdral (2019)	-	0.946	0.942	0.949	0.954	-	0.733	0.641	0.803	0.877
Rotate3D (2020)	214	0.951	**0.945**	0.953	0.961	39	0.789	0.728	0.832	0.887
QuatE (2019)	338	0.949	0.941	0.954	0.960	41	0.770	0.700	0.821	0.878
DualE (2021)	-	0.951	**0.945**	**0.956**	0.961	-	0.790	0.734	0.829	0.881
RotateCT (2022)	201	0.951	0.944	**0.956**	0.963	**34**	0.794	0.737	0.834	0.888
DCNE (ours)	**192**	**0.952**	**0.945**	0.955	0.963	**34**	**0.798**	0.745	**0.835**	0.888

Table 1: Link prediction results on WN18 and FB15k.

Model	**Rotate3D**	**DCNE**	**QuatE**
Space	$\mathbb{H}^k$	$\mathbb{DC}^k$	$\mathbb{H}^k$
Dimension	1000	1000	1000
FB15k	50.23M	33.94M(↓ 32.4%)	69.18M
FB15k-237	44.56M	29.79M(↓ 33.1%)	59.11M
WN18	122.90M	81.94M(↓ 33.3%)	163.84M
WN18RR	122.87M	81.92M(↓ 33.3%)	163.82M
Freebase	225.18M	150.13M(↓ 33.3%)	300.22M
WordNet	115.70M	77.14M(↓ 33.3%)	154.25M

Table 2: Number of free parameters.

TorusE (Ebisu and Ichise, 2018), RotatE (Sun et al., 2019), BoxE (Abboud et al., 2020), MuRP (Balazevic et al., 2019a) and Rot-Pro (Song et al., 2021); for models with non-commutativity, we report DihEdral (Xu and Li, 2019), QuatE (Zhang et al., 2019), Rotate3D (Gao et al., 2020), DualE (Cao et al., 2021) and RotateCT (Dong et al., 2022); for indeterminate models, we report ConvE (Dettmers et al., 2018), R-GCN+ (Schlichtkrull et al., 2018) and NKGE (Wang et al., 2018).

The empirical results on four benchmarks are reported in Table 1 and Table 3. Best results are in bold and second best results are underlined. We can see that DCNE outperforms all the baselines on WN18, WN18RR and FB15k-237, and achieves extremely competitive performance on FB15k, which demonstrates the effectiveness of DCNE in single-hop reasoning. On WN18RR and FB15k-237, the main relation patterns are symmetry/antisymmetry and composition. The performance improvement confirms the effectiveness of DCNE in modeling symmetry/antisymmetry and composition patterns. As a dual-complex-number-valued method, DCNE outperforms two representative quaternion-valued methods QuatE and Rotate3D, the dual-quaternion-valued method DualE, and the complex-valued method RotateCT, which fully demonstrates the superiority of dual complex number embeddings. The overall performance of the models with non-commutativity is better than that of the models without non-commutativity, which proves that modeling non-commutative composition patterns is crucial. On WN18 and FB15k, the main relation patterns are symmetry/antisymmetry and inversion. Since DCNE has no obvious superiority over other state-of-the-art baselines in modeling symmetry/antisymmetry and inversion patterns, the performance improvement on WN18 and FB15k is not significant than that on WN18RR and FB15k-237.

Ablation study. We conduct ablation study of self-adversarial negative sampling on WN18RR and FB15k-237. From Table 4, we observe that the performance of DCNE drops by an average of 0.475 percentage points across MRR, Hits@1, Hits@3 and Hits@10 on FB15k-237 when not using self-adversarial negative sampling, where RotatE drops by an average of 4.425 percentage points and RotateCT (Dong et al., 2022) drops by an average of 3.175 percentage points. One reason is that the performance improvement of our model comes from dual complex number embeddings, not self-adversarial negative sampling, which is also confirmed by the ablation results of DCNE on WN18RR.

5.2. Path Query Answering

This task is to answer path queries on KGs (Guu et al., 2015). Given a path query q consisting of a start entity s and a path p, the answer of q is

Model	WN18RR					FB15k-237				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TransE (2013)	3384	0.226	-	-	0.501	357	0.294	-	-	0.465
DistMult (2015)	5100	0.430	0.390	0.440	0.490	254	0.241	0.155	0.263	0.419
ComplEx (2016)	5261	0.440	0.410	0.460	0.510	339	0.247	0.158	0.275	0.428
MuRP (2019a)	-	0.475	0.436	0.487	0.554	-	0.336	0.245	0.370	0.521
BoxE (2020)	3207	0.451	0.400	0.472	0.541	**163**	0.337	0.238	0.374	0.538
RotatE † (2019)	3340	0.476	0.428	0.492	0.571	177	0.338	0.241	0.375	0.533
Rot-Pro † (2021)	**2815**	0.457	0.397	0.482	0.577	201	0.344	0.246	0.383	0.540
ConvE (2018)	4187	0.430	0.400	0.440	0.520	244	0.325	0.237	0.356	0.501
R-GCN+ (2018)	-	-	-	-	-	-	0.249	0.151	0.264	0.417
NKGE (2018)	4170	0.450	0.421	0.465	0.526	237	0.330	0.241	0.365	0.510
DihEdral (2019)	-	0.486	0.442	0.505	0.557	-	0.320	0.230	0.353	0.502
Rotate3D † (2020)	3328	0.489	0.442	0.505	0.579	165	0.347	0.250	0.385	0.543
QuatE (2019)	3472	0.481	0.436	0.500	0.564	176	0.311	0.221	0.342	0.495
DualE (2021)	-	0.482	0.440	0.500	0.561	-	0.330	0.237	0.363	0.518
RotateCT (2022)	3285	**0.492**	**0.448**	0.507	0.579	171	0.347	0.251	0.382	0.537
DCNE (ours)	3244	**0.492**	**0.448**	**0.510**	**0.581**	169	**0.354**	**0.257**	**0.393**	**0.547**

Table 3: Link prediction results on WN18RR and FB15k-237. [†]: Results with the self-adversarial negative sampling.

	WN18RR					**FB15k-237**				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
RotatE w/ self-adv ♯	3340	0.476	0.428	0.492	0.571	177	0.338	0.241	0.375	0.533
RotatE w/o self-adv ♯	-	-	-	-	-	185	0.297	0.205	0.328	0.480
RotateCT w/ self-adv ♯	3285	**0.492**	**0.448**	0.507	0.579	171	0.347	0.251	0.382	0.537
RotateCT w/o self-adv	2950	0.486	0.441	0.501	0.572	175	0.314	0.215	0.352	0.509
DCNE w/ self-adv	3244	**0.492**	**0.448**	**0.510**	**0.581**	**169**	**0.354**	**0.257**	**0.393**	**0.547**
DCNE w/o self-adv	**2908**	0.489	0.446	0.504	0.574	181	0.349	0.253	0.388	0.542

Table 4: Ablation study results on WN18RR and FB15k-237, where "self-adv" denotes the self-adversarial negative sampling. [♯]: Results are taken from the original paper.

the entities that can be reached from s via p. A path p consists of a sequence of relations, i.e., $\mathrm{r}_1 \rightarrow ... \rightarrow \mathrm{r}_\mathrm{j}$, where j is the length of p. Notably, link prediction can be viewed as a special case of path query answering when the path length is fixed to 1. However, different from link prediction, path query answering requires the ability of mutihop reasoning, in which modeling the composition patterns is crucial.

Datasets. We conduct experiments on two datasets released by Guu et al. (2015), which are extracted from WordNet and Freebase. Both datasets contain triplets and paths.

Evaluation Protocol. We use the same evaluation protocol as in (Guu et al., 2015). For each test path $\mathrm{p_t} = \mathrm{s} \rightarrow \mathrm{r}_1 \rightarrow ... \rightarrow \mathrm{r}_\mathrm{j} \rightarrow \mathrm{o}$, the corresponding query q is $\mathrm{s} \rightarrow \mathrm{r}_1 \rightarrow ... \rightarrow \mathrm{r}_\mathrm{j} \rightarrow ?$. For each query q, the candidate answers are entities that "type-match", i.e., all tail entities of the final relation r_j. The correct answers are entities that can be reached from s by traversing the path p; the incorrect answers are obtained by filtering out the correct answers from the candidate answers. The set of candidate answers to a query q denotes as $\mathcal{C}(\mathrm{q})$; the set of correct answers to a query q denotes as $\mathcal{P}(\mathrm{q})$; the set of incorrect answers to a query q denotes as $\mathcal{N}(\mathrm{q})$. We give the formal definition of $\mathcal{C}(\mathrm{q})$, $\mathcal{P}(\mathrm{q})$, and $\mathcal{N}(\mathrm{q})$ as follows:

$$\mathcal{C}(\mathrm{q}) \triangleq \{\mathrm{o} | \exists \mathrm{e} \text{ s.t. } (\mathrm{e}, \mathrm{r_j}, \mathrm{o}) \in \mathcal{G}\}$$
$$\mathcal{P}(\mathrm{q}) \triangleq \{\mathrm{o} | \exists \mathrm{e}_1, ..., \mathrm{e}_{\mathrm{j}-1} \text{ s.t. } (\mathrm{s}, \mathrm{r}_1, \mathrm{e}_1), ..., (\mathrm{e}_{\mathrm{j}-1}, \mathrm{r_j}, \mathrm{o}) \in \mathcal{G}\}$$
$$\mathcal{N}(\mathrm{q}) \triangleq \mathcal{C}(\mathrm{q}) \backslash \mathcal{P}(\mathrm{q})$$

Here $\mathcal{G}$ includes training triplets and test triplets. For each test path $\mathrm{p_t}$, we replace entity o with entities in $\mathcal{C}(\mathrm{q})$ and compute the score of each candidate answer. Then we rank the scores of candidates along with the score of $\mathrm{p_t}$ in descending order and caculate the quantile, which is the proportion of incorrect answers ranked after the target answer o. We report the mean quantile (MQ) and Hits@10 on this task. MQ is the average quantile of all test paths. Hits@10 represents the percentage of target answers whose rank is not larger than 10. Higher MQ and higher Hits@10 indicate the better performance.

Model	WordNet MQ	WordNet Hits@10	Freebase MQ	Freebase Hits@10
DistMult	0.904	0.311	0.848	0.386
TransE	0.933	0.435	0.880	0.505
RotatE	0.947	0.653	0.901	0.601
Rotate3D	**0.949**	0.671	0.905	0.621
RotateCT	**0.949**	0.673	0.907	**0.630**
DCNE (ours)	**0.949**	**0.674**	**0.911**	**0.630**

Table 5: Path query answering results on WordNet and Freebase.

Implementation Details. We train DCNE with all paths in the training set, which is denoted as "Comp" in (Guu et al., 2015). Note that triplets are the paths of length 1. To make the results directly comparable, we follow Gao et al. (2020) to train DCNE on paths of length 1 to 5 in turn, i.e., we train our model on paths of length i until convergence before training on paths of length $i+1$. The experimental environment is the same as in link prediction.

Main Results. We compare our method DCNE to several representative models, such as DistMult (Yang et al., 2015), TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), Rotate3D (Gao et al., 2020) and RotateCT (Dong et al., 2022).

Table 5 shows experimental results on both WordNet and Freebase. Best results are in bold and second best results are underlined. Compared with other methods, DCNE achieves consistent performance improvement on the two datasets, which indicates the superiority of DCNE in modeling the composition patterns. Specifically, DCNE significantly outperforms DistMult, TransE and RotatE, while slightly surpasses Rotate3D. We argue that this is because DCNE and Rotate3D both have the ability of modeling the non-commutative composition patterns, whereas others do not.

6. Case Studies

In this section, we verify that DCNE can effectively model all the three types of relation patterns by histograms, some intuitive examples and quantitative experiments.

6.1. Symmetry/Antisymmetry

According to Corollary 1, the phase of each element in the embeddings of a symmetric relation should be 0 or $\pm\pi$. For an antisymmetric relation, the phases should not be 0 and $\pm\pi$. We investigate the phases of elements in relation embeddings from a 1000-dimensional DCNE trained on WN18 and a 1000-dimensional DCNE trained on FB15k-237 with their best hyperparameters in link prediction.

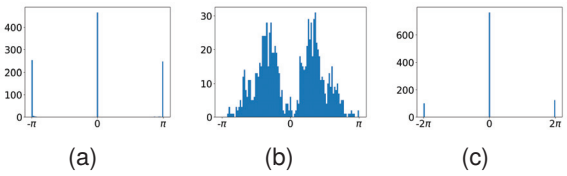


Figure 3: (a) Histogram of relation embeddings phases $\{\theta_{r,i}\}(r_i = cos\frac{\theta_{r,i}}{2} + sin\frac{\theta_{r,i}}{2}(i + x_{r,i}\epsilon + y_{r,i}i\epsilon))$ of a symmetric relation: $\mathrm{similar\_to}$. (b) Histogram of relation embeddings phases $\{\theta_{r,i}\}$ of an antisymmetric relation, where $\mathrm{parent\_genre}$ represents $/\mathrm{music/genre/parent\_genre}$. (c) Histogram of the addition of relation embeddings phases $\{(\theta_{1,i}+\theta_{2,i})\}$ of a pair of inverse relations: $\mathrm{has\_part} \circ \mathrm{part\_of}$, where $\circ$ is the Hadmard (or element-wise) product.

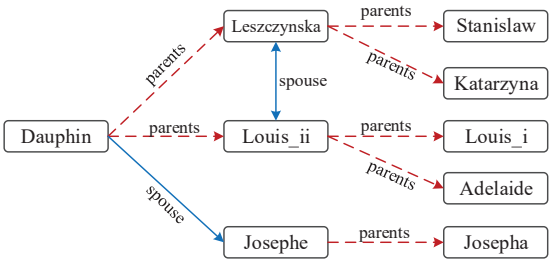


Figure 4: A subgraph about Dauphin's family. Dauphin's parents' spouse are Leszczynska and Louis_ii; Dauphin's spouse's parents is Josepha; Dauphin's parents' parents are Stanislaw, Katarzyna, Louis_i and Adelaide.

Fig. 3(a) gives the histogram of a symmetric relation in WN18: $\mathrm{similar\_to}$. We can find that most phases of $\mathrm{similar\_to}$ are either 0 or $\pm\pi$. The histogram of an antisymmetric relation $\mathrm{parent\_genre}$ in FB15k-237 is shown in Fig. 3(b), from which we can find that the phases of $\mathrm{parent\_genre}$ are scattered and most phases of $\mathrm{parent\_genre}$ are neither 0 nor $\pm\pi$. The above observations indicate that DCNE can effectively model the symmetry/antisymmetry patterns.

6.2. Inversion

According to Corollary 2, if there exists inverse relationship between r_1 and r_2, each element of the additive embedding phases, i.e., $\theta_{1,i}+\theta_{2,i}$, should be 0 or $\pm 2\pi$. We use the same DCNE model trained on WN18 for an analysis. Fig. 3(c) shows the element-wise addition of the embedding phases of a pair of inverse relations: $\mathrm{has\_part}$ and $\mathrm{part\_of}$. We can find that most additive embedding phases are either 0 or $\pm 2\pi$, which illustrates that DCNE can effectively model the inversion patterns.

6.3. Composition

Fig. 4 shows a subgraph extracted from the Freebase dataset. In this subgraph, $\mathrm{parents}$ and spouse form a non-commutative composition pattern, in

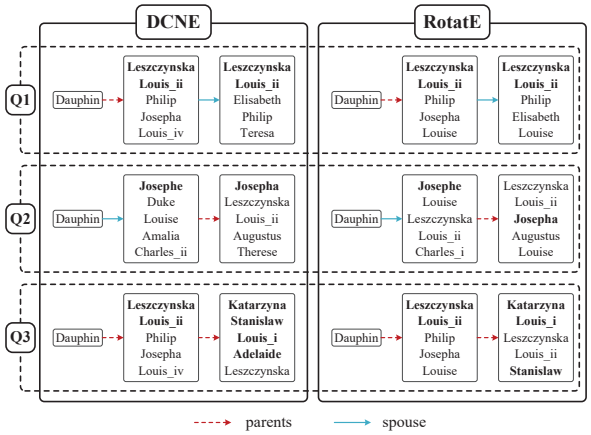


Figure 5: Top 5 answers of DCNE and RotatE for three queries. Q1: Who are Dauphin's parents' spouse? Q2: Who are Dauphin's spouse's parents? Q3: Who are Dauphin's parents' parents? Correct answers are in **bold**.

	Hits@1			Hits@3		
	RotatE	Rotate3D	DCNE	RotatE	Rotate3D	DCNE
parents' spouse	0.920	0.971	**0.973**	0.997	**1.000**	**1.000**
spouse's parents	0.692	**0.806**	0.804	0.922	0.990	**0.992**
parents' parents	0.910	0.967	**0.968**	0.984	**1.000**	**1.000**

Table 6: Results of two non-commutative composition patterns "parents' spouse" and "spouse's parents", and one commutative composition pattern "parents' parents" on the Freebase dataset.

which the meaning of "parents' spouse" is different from "spouse's parents". In addition, there is a commutative composition pattern "parents' parents". We use a DCNE and a RotatE, both trained with their best hyperparameters on the Freebase in path query answering for an analysis.

Non-commutative Composition. From Fig. 5, we find that DCNE predicts the queries "Q1: Who are Dauphin's parents' spouse?" and "Q2: Who are Dauphin's spouse's parents?" correctly, which indicates that DCNE can effectively model the non-commutative composition patterns. By comparision, RotatE confuses the two queries and gives unsatisfactory answers to the query "Q2: Who are Dauphin's spouse's parents?", which proves that RotatE lacks the ability of modeling the non-commutative composition patterns.

Commutative Composition. We further investigate the ability of modeling the commutative composition patterns. As shown in Fig. 5, DCNE predicts the query "Q3: Who are Dauphin's parents' parents?" correctly, which indicates that DCNE can effectively model the commutative composition patterns. By contrast, RotatE misses one correct answer in top 5 answers. One reason is that the inability of RotatE in modeling the non-commutative composition patterns has an adverse impact on embeddings' learning. Intuitively, if RotatE represents "parents' spouse" and "spouse's parents" in the same way, it is hard to decide what semantic information to retain and what to discard. This dilemma prevents RotatE from effectively learning the semantics of parents and spouse, which in turn, will reduce RotatE's performance on modeling the pattern "parents' parents".

To verify our analysis, we evaluate two non-commutative composition patterns "parents' spouse" and "spouse's parents", and one commutative composition pattern "parents' parents" on the Freebase dataset. From Table 6, we observe that DCNE outperforms RotatE on all three patterns and obtains better performance than Rotate3D, which fully demonstrates the effectiveness of dual complex number embeddings in modeling the non-commutative composition patterns. Due to effectively learning the semantics of parents and spouse, DCNE achieves better performance than RotatE on the "parents' parents" pattern.

7. Conclusion

In this paper, we propose a new KGE method called DCNE, which projects entities to the dual complex number space, and represents relations as rotations around any point in 2-dimensional Euclidean space via dual complex number multiplication. Due to the non-commutativity of the dual complex number multiplication, DCNE can model not only the commutative composition patterns, but also the non-commutative composition patterns. In the meantime, modeling relations as rotations in 2D space can effectively improve the parameter efficiency. Experimental results on link prediction and path query answering demonstrate the superiority of DCNE in single-hop reasoning and multi-hop reasoning. Case studies confirm the ability of DCNE in modeling all the relation patterns. Parameter efficiency analysis proves that DCNE can reduce the space cost compared to quaternion-valued methods. In future work, we will explore the different usages of dual complex numbers.

8. Acknowledgements

This work was supported in part by the Ministry of Science and Technology of China under Grant #2020AAA0108405, and in part by the National Natural Science Foundation of China under Grants #62206287, the Beijing Nova Program Z201100006820085 from Beijing Municipal Science and Technology Commission.

9. Bibliographical References

Ralph Abboud, Ismail Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. 2020. BoxE: A box embedding model for knowledge base completion. In *NeurIPS*, pages 9649–9661.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019a. Multi-relational poincaré graph embeddings. In *NeurIPS*, pages 4463–4473.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019b. TuckER: Tensor factorization for knowledge graph completion. In *EMNLP*, pages 5185–5194.

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

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

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2021. Dual quaternion knowledge graph embeddings. In *AAAI*, 8, pages 6894–6902.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *ACL*, pages 6901–6914.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2017. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *EACL*, pages 132–141.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *AAAI*, 1, page 1811–1818.

Yao Dong, Lei Wang, Ji Xiang, Xiaobo Guo, and Yuqiang Xie. 2022. Rotatect: Knowledge graph embedding by rotation and coordinate transformation in complex space. In *COLING*, pages 4918–4932.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: Knowledge graph embedding on a lie group. In *AAAI*, 1, pages 1819–1826.

Chang Gao, Chengjie Sun, Lili Shan, Lei Lin, and Mingjiang Wang. 2020. Rotate3d: Representing relations as rotations in three-dimensional space for knowledge graph embedding. In *CIKM*, pages 385–394.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *EMNLP*, pages 318–327.

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 *ACL*, pages 221–231.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, pages 687–696.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *NeurIPS*, pages 4284–4295.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Rui Li, Jianan Zhao, Chaozhuo Li, Di He, Yiqi Wang, Yuming Liu, Hao Sun, Senzhang Wang, Weiwei Deng, Yanming Shen, Xing Xie, and Qi Zhang. 2022. HousE: Knowledge graph embedding with householder parameterization. In *ICML*, pages 13209–13224.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, 1, pages 2181–2187.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119.

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

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, et al. 2018. Never-ending learning. *Communications of the ACM*, 61(5):103–115.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *ACL*, pages 4710–4723.

Tu Dinh Nguyen, Dat Quoc Nguyen, Dinh Phung, et al. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *NAACL*, pages 327–333.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *AAAI*, 1, pages 1955–1961.

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

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

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

Tengwei Song, Jie Luo, and Lei Huang. 2021. Rotpro: Modeling transitivity by projection in knowledge graph embedding. In *NeurIPS*, pages 24695–24706.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*, pages 697–706.

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

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*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*, pages 2071—2080.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. In *EMNLP*, pages 1257–1266.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha P Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *AAAI*, pages 3009–3016.

Kai Wang, Yu Liu, Xiujuan Xu, and Dan Lin. 2018. Knowledge graph embedding with entity neighbors and deep memory network. *arXiv preprint arXiv:1808.03752*.

Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019. Coke: Contextualized knowledge graph embedding. *arXiv preprint arXiv:1911.02168*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 1, page 1112–1119.

Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016. From one point to a manifold: knowledge graph embedding for precise link prediction. In *IJCAI*, pages 1315–1321.

Canran Xu and Ruijiang Li. 2019. Relation embedding with dihedral group in knowledge graph. In *ACL*, pages 263–272.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Wenpeng Yin, Yadollah Yaghoobzadeh, and Hinrich Schütze. 2018. Recurrent one-hop predictions for reasoning over knowledge graphs. In *COLING*, pages 2369–2378.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*, pages 353–362.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *NeurIPS*, pages 2735–2745.

Yang Zhao, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2020. Knowledge graphs enhanced neural machine translation. In *IJCAI*, pages 4039–4045.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.