TransERR: Translation-based Knowledge Graph Embedding via Efficient Relation Rotation

Jiang Li[1,2,3]**, Xiangdong Su**[1,2,3(✉)]**, Fujun Zhang**[1,2,3]**, Guanglai Gao**[1,2,3]

[1]College of Computer Science, Inner Mongolia University, Hohhot, China
[2]National & Local Joint Engineering Research Center of
Intelligent Information Processing Technology for Mongolian
[3]Inner Mongolia Key Laboratory of Mongolian Information Processing Technology
lijiangimu@gmail.com, cssxd@imu.edu.cn,32209093@mail.imu.edu.cn, csggl@imu.edu.cn

Abstract

This paper presents a translation-based knowledge geraph embedding method via efficient relation rotation (TransERR), a straightforward yet effective alternative to traditional translation-based knowledge graph embedding models. Different from the previous translation-based models, TransERR encodes knowledge graphs in the hypercomplex-valued space, thus enabling it to possess a higher degree of translation freedom in mining latent information between the head and tail entities. To further minimize the translation distance, TransERR adaptively rotates the head entity and the tail entity with their corresponding unit quaternions, which are learnable in model training. We also provide mathematical proofs to demonstrate the ability of TransERR in modeling various relation patterns, including symmetry, antisymmetry, inversion, composition, and subrelation patterns. The experiments on 10 benchmark datasets validate the effectiveness and the generalization of TransERR. The results also indicate that TransERR can better encode large-scale datasets with fewer parameters than the previous translation-based models. Our code and datasets are available at `https://github.com/dellixx/TransERR`.

Keywords: knowledge graph embedding, link prediction, knowledge graph

1. Introduction

Knowledge graphs (KGs), also known as semantic networks, represent networks of real-world entities (objects, events, concepts, etc.) and describe the associations between them. In fact, KGs contain factual triplets $(head, relation, tail)$, which are denoted as (h, r, t). Several open source KGs are available, including FreeBase (Bollacker et al., 2008), DBpedia (Lehmann et al., 2015) and NELL (Mitchell et al., 2018). They facilitate the development of downstream tasks, such as question answering (Chen et al., 2019), semantic search (Junior et al., 2020) and relation extraction (Hu et al., 2021). However, there is a problem with missing links in KGs. Therefore, knowledge graph embedding (KGE) task has recently received considerable attention.

The mainstream approaches are to learn low-dimensional representations of entities and relations and utilize them to predict new facts. Most of them learn the embeddings of KGs based on score functions of the translational distance between the head and tail entities. Weak degree of freedom (✗) and Translational Transformations (✔). RotatE (Sun et al., 2019), PairRE (Chao et al., 2021) and TranSHER (Li et al., 2022b) have demonstrated that translational transformations better capture the relations between entities, thereby enhancing the expressive power of KG embeddings. However, these models are constrained by the degrees of freedom in the transformations, implying that they are not fully expressive.

Recently, QuatE (Zhang et al., 2019) and Rotat3D (Gao et al., 2020) have demonstrated that vectors exhibit a higher degree of rotational freedom in the quaternion space. This implies that one can attain a greater level of rotational freedom in representing when compared to traditional vector spaces. Nevertheless, QuatE and Rotat3D exclusively employ rotational transformations within quaternion space, thus overlooking translational changes. This limitation leads to a reduced capacity for modeling spatial transformations, resulting in a decrease in performance of link prediction. High degree of freedom (✔) and Translational Transformations (✗).

Based on the above facts, this paper proposes a KGE method named TransERR that combines a high degree of rotational freedom (✔) and translational transformations (✔). TransERR encodes knowledge graphs in the hypercomplex-valued space and utilizes two unit quaternion vectors to rotate the head entity and the tail entity, respectively. The unit quaternion vectors are learnable in model training, and they are highly to smooth rotation and spatial translation in the hypercomplex-valued vector space. In addition, two unit quaternion rotation vectors can further narrow the translation distance between the head and tail entities. As a result, TransERR possesses a higher degree of translation freedom in mining latent information between the head and tail entities. We evaluate the effectiveness and generalisation of our model on 10 KG

benchmark datasets of different sizes. The experimental results show that TransERR significantly outperforms the existing state-of-the-art distance-based models. We provide mathematical proofs to demonstrate that TransERR can infer key relation patterns simultaneously. Furthermore, we demonstrate that TransERR can better model complex relations of KGs than the existing distance-based models and surpass the baselines on large-scale datasets with fewer parameters.

2. Related Work

Most existing KGE models can be roughly classified into two categories: Translation Models and Semantic Matching Models (Wang et al., 2017; Ji et al., 2021). The former measures the plausibility of a fact as a translation distance between two entities, while the latter measures the plausibility of facts by matching latent semantics of entities and relations. In this paper, TransERR belongs to translation models.

Translation Models. Translation-based models, also known as distance-based models. Motivated by the translation invariance in word2vec (Mikolov et al., 2013), TransE (Bordes et al., 2013) defines the distance between $h + r$ and t with the l_1 or l_2 norm constraint. After that, TransH (Wang et al., 2014), TransR (Lin et al., 2015) and TransD (Ji et al., 2015) employ different projection strategies to adjust graph embeddings. TranSparse (Ji et al., 2016) overcomes heterogeneity and imbalance by combining TranSparse (share) and TranSparse (separate). TransMS (Yang et al., 2019) perform multi-directional semantic transfer with non-linear functions and linear deviation vectors. RotatE (Sun et al., 2019) defines each relation as a rotation for a triplet from the head entity to the tail entity. Rotat3D (Gao et al., 2020) maps entities into 3D space and defines relations similar to RotatE. However, Rotat3D is limited by optimizing the translation distance with only relation embedding information. Recently, PairRE (Chao et al., 2021), TripleRE (Yu et al., 2022) and TranSHER (Li et al., 2022b) employ multiple relations to improve the degree of freedom for relational rotation. Different from previous work, TransERR processes a high degree of rotation in the quaternion space. Benefiting from our model structure, TransERR can model all important relation patterns simultaneously and allow for better optimization of the translation distance between entities and, thus better mine the latent information.

Semantic Matching Models. Semantic matching models mine possible semantic associations between entities and relations. The RESCAL (Nickel et al., 2011) represents each relation as a non-singular matrix. DistMult (Yang et al., 2014) uses a diagonal matrix rather than a non-singular matrix to address the problem of the excessive number of parameters in RESCAL. ComplEx (Trouillon et al., 2016) introduces complex-valued spaces into the KGs. TuckER (Balažević et al., 2019) employs Tucker decomposition of a binary tensor to model a KG. SimplE$^+$ (Fatemi et al., 2019) extends SimplE (Kazemi and Poole, 2018) and focuses on encoding the subrelation pattern. QuatE (Zhang et al., 2019), QuatRE (Nguyen et al., 2022) and QuatSE (Li et al., 2022a) take advantage of quaternion representations to enable rich interactions between entities and relations. Deep neural networks have received a great deal of attention in recent years, e.g., ConvE (Dettmers et al., 2018a), ConvKB (Nguyen et al., 2018), InteractE (Vashishth et al., 2020) and R-GCNs (Schlichtkrull et al., 2018). However, they are difficult to analyze as they work as a black box.

3. Background and Notation

In this section, we first introduce the link prediction task. Secondly, we introduce the key relation patterns and complex relation patterns that are commonly studied in this task. Finally, we provide a brief introduction to quaternion algebra.

Link Prediction. A knowledge graph is usually described as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ denote the set of entities, relations and triplets (h, r, t), respectively. Specifically, given $(h, r, ?)$, link prediction is to predict the tail entity in the triplet. Alternatively, given $(?, r, t)$, the task is to predict the head entity. The existing predicting models calculate the score function $f_r(h, t)$ and expect that the scores of correct triplets are higher than those of invalid triplets.

Key Relation Patterns. In this part, we introduce several important relation patterns that have been extensively studied in link prediction task. Following Fatemi et al. (2019) and Sun et al. (2019), relations can also be summarized by several patterns:

- **Symmetry** If $(e_1, r, e_2) \in \mathcal{T}, \forall e_1, e_2 \in \mathcal{E} \Rightarrow (e_2, r, e_1) \in \mathcal{T}$.

- **Antisymmetry** If $(e_1, r, e_2) \in \mathcal{T}, \forall e_1, e_2 \in \mathcal{E} \Rightarrow (e_2, r, e_1) \notin \mathcal{T}$.

- **Inversion** If $(e_1, r_1, e_2) \in \mathcal{T}, \forall e_1, e_2 \in \mathcal{E} \Rightarrow (e_2, r_2, e_1) \in \mathcal{T}$.

- **Composition** If $(e_1, r_1, e_2) \in \mathcal{T} \wedge (e_2, r_2, e_3) \in \mathcal{T}, \forall e_1, e_2, e_3 \in \mathcal{E} \Rightarrow (e_1, r_3, e_3) \in \mathcal{T}$.

- **Subrelation** If $(e_1, r_1, e_2) \in \mathcal{T}, \forall e_1, e_2 \in \mathcal{E} \Rightarrow (e_1, r_2, e_2) \in \mathcal{T}$. $(e_1, r_1, e_2) \to (e_1, r_2, e_2)$ and r_2 can be considered as a subrelation of r_1.

Complex Relation Patterns. We take the definition of complex relations from Wang et al. (2014),
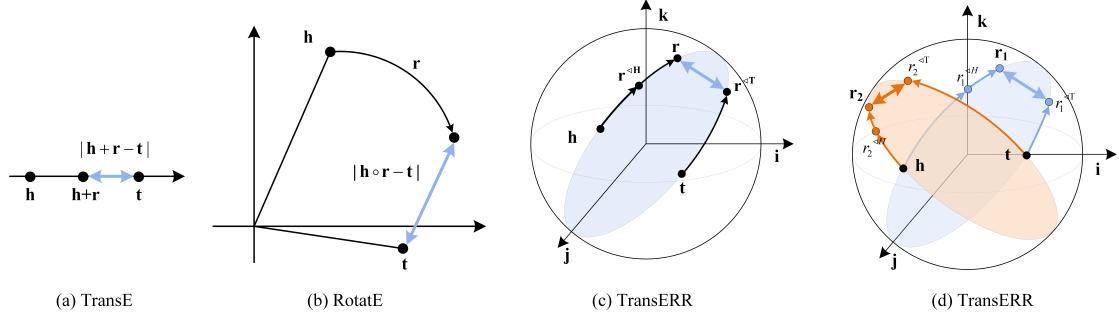
Figure 1: Illustration of TransE, RotatE and TransERR. TransE, RotatE and TransERR encode knowledge graphs in the real-valued space, complex-valued space and hypercomplex-valued space, respectively. ∘ denotes Hadamard product. The distance function of TransERR is $\| \mathbf{h} \otimes \mathbf{r}^{\triangleleft\mathbf{H}} + \mathbf{r} - \mathbf{t} \otimes \mathbf{r}^{\triangleleft\mathbf{T}} \|$.

and we calculate average number of tails per head (tph_r) and average number of heads per tail (hpt_r). If $tph_r < 1.5$ and $hpt_r < 1.5$, r is defined as one-to-one (**1-1**). If $tph_r > 1.5$ and $hpt_r > 1.5$, r is defined as many-to-many (**N-N**). If $tph_r > 1.5$ and $hpt_r < 1.5$, r is defined as one-to-many (**1-N**). If $tph_r < 1.5$ and $hpt_r > 1.5$, r is defined as many-to-one (**N-1**).

Quaternion Algebra. The quaternion is an extension of the complex number in the four-dimensional space. It consists of a real part and three imaginary parts, which is proposed by William Rowan Hamilton (Hamilton, 1844). A quaternion q is defined as $q = a + bi + cj + dk$, where a is real unit and i, j, k are three imaginary units. In addition, $i^2 = j^2 = k^2 = ijk = -1$.

- The unit quaternion $q^{\triangleleft}$ is defined as

$$q^{\triangleleft} = \frac{a + bi + cj + dk}{\sqrt{a^2 + b^2 + c^2 + d^2}}. \tag{1}$$

- Hamilton product of two quaternions is

$$\begin{aligned} q_1 \otimes q_2 =& (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) + \\ & (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)i + \\ & (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)j + \\ & (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)k. \end{aligned} \tag{2}$$

4. Methodology

4.1. TransERR

Quaternions enable expressive rotation in the hypercomplex-valued space and have more degree of freedom than translation in the real-valued space. Hence, to obtain a greater degree of translation freedom in the link prediction, TransERR encodes knowledge graphs in the hypercomplex-valued space. In addition, we rotate head entity $\mathbf{h}$ and tail entity $\mathbf{t}$ via two unit quaternion vectors $\mathbf{r}^{\triangleleft\mathbf{H}}$ and $\mathbf{r}^{\triangleleft\mathbf{T}}$, as shown in Figure 1. Unlike TransE and RotatE, TransERR utilizes Hamilton product

$\otimes$ rather than Hadamard product $\circ$ in project operation to better capture the underlying semantic features between the head entity and the tail entity embeddings. Firstly, given a triplet (h, r, t), we encode h, r and t in the quaternion space, which are denoted as

$$\begin{aligned} \mathbf{h} &= \mathbf{a_h} + \mathbf{b_h}i + \mathbf{c_h}j + \mathbf{d_h}k \\ \mathbf{r} &= \mathbf{a_r} + \mathbf{b_r}i + \mathbf{c_r}j + \mathbf{d_r}k \\ \mathbf{t} &= \mathbf{a_t} + \mathbf{b_t}i + \mathbf{c_t}j + \mathbf{d_t}k, \end{aligned} \tag{3}$$

where $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{H}^d$, $\mathbf{a_h}, \mathbf{b_h}, \mathbf{c_h}, \mathbf{d_h} \in \mathbb{R}^{\frac{d}{4}}$, $\mathbf{a_r}, \mathbf{b_r}, \mathbf{c_r}, \mathbf{d_r} \in \mathbb{R}^{\frac{d}{4}}$ and $\mathbf{a_t}, \mathbf{b_t}, \mathbf{c_t}, \mathbf{d_t} \in \mathbb{R}^{\frac{d}{4}}$. Next, we define two additional quaternion vectors to rotate the head and tail entity, which are represented as $\mathbf{r^H}$ and $\mathbf{r^T}$, where $\mathbf{r^H}$ and $\mathbf{r^T} \in \mathbb{H}^d$. Then, we normalize these two additional quaternions ($\mathbf{r^H}$ and $\mathbf{r^T}$) to the unit quaternions ($\mathbf{r}^{\triangleleft\mathbf{H}}$ and $\mathbf{r}^{\triangleleft\mathbf{T}}$) to eliminate the scaling effect. They are denoted as

$$\begin{aligned} \mathbf{r}^{\triangleleft\mathbf{H}} &= \frac{\mathbf{a_{r^H}} + \mathbf{b_{r^H}}i + \mathbf{c_{r^H}}j + \mathbf{d_{r^H}}k}{\sqrt{\mathbf{a_{r^H}^2} + \mathbf{b_{r^H}^2} + \mathbf{c_{r^H}^2} + \mathbf{d_{r^H}^2}}} \\ \mathbf{r}^{\triangleleft\mathbf{T}} &= \frac{\mathbf{a_{r^T}} + \mathbf{b_{r^T}}i + \mathbf{c_{r^T}}j + \mathbf{d_{r^T}}k}{\sqrt{\mathbf{a_{r^T}^2} + \mathbf{b_{r^T}^2} + \mathbf{c_{r^T}^2} + \mathbf{d_{r^T}^2}}}. \end{aligned} \tag{4}$$

The normalized operation ensures the stability of entity rotation in the quaternion space. The unit quaternion vectors are highly desirable to smooth rotation and spatial translation in the hypercomplex-valued space. In addition, two unit quaternion rotation vectors can further narrow the translation distance between the head and tail entities. Finally, we employ $\mathbf{r}^{\triangleleft\mathbf{H}}$ and $\mathbf{r}^{\triangleleft\mathbf{T}}$ to rotate head entity $\mathbf{h}$ and tail entity $\mathbf{t}$, respectively. Specifically, we use Hamilton product to achieve rotation operation since Hamilton product makes quaternion more expressive at rotational capability. For each triplet (h, r, t), we define the distance function of TransERR as

$$d_r(\mathbf{h}, \mathbf{t}) = \| \mathbf{h} \otimes \mathbf{r}^{\triangleleft\mathbf{H}} + \mathbf{r} - \mathbf{t} \otimes \mathbf{r}^{\triangleleft\mathbf{T}} \|. \tag{5}$$

The score function $f_r(\mathbf{h}, \mathbf{t}) = -d_r(\mathbf{h}, \mathbf{t})$ and $\otimes$ is defined in Section 3. Following Sun et al. (2019),

we employ the self-adversarial negative sampling loss for TransERR, which is defined as

$$\mathcal{L} = -\log \sigma(\gamma - d_r(\mathbf{h}, \mathbf{t}))$$
$$- \sum_{i=1}^{n} p(h_i', r, t_i') \log \sigma(d_r(\mathbf{h_i'}, \mathbf{t_i'}) - \gamma), \quad (6)$$

where σ is the sigmoid function, and γ is a fixed margin. (h_i', r, t_i') and $d_r(\mathbf{h_i'}, \mathbf{t_i'})$ represent the i-th negative triplet and the distance function of the i-th negative triplet. The weight of the negative sample $p(h_i', r, t_i')$ is defined as

$$p((h_i', r, t_i')|(h, r, t)) = \frac{\exp \alpha f_r(\mathbf{h_i'}, \mathbf{t_i'})}{\sum_j \exp \alpha f_r(\mathbf{h_j'}, \mathbf{t_j'})}, \quad (7)$$

where α indicates the temperature of sampling.

4.2. Theoretical Analysis

TransERR can model important relation patterns, including symmetry, antisymmetry, inversion, composition, subrelation and multiple. Please refer to Appendix A for a detailed proof process.

Proposition 1. *TransERR can infer the symmetry relation pattern.*

Proposition 2. *TransERR can infer the antisymmetry relation pattern.*

Proposition 3. *TransERR can infer the inversion relation pattern.*

Proposition 4. *TransERR can infer the composition relation pattern.*

Proposition 5. *TransERR can infer the subrelation pattern.*

Dataset	$|\mathcal{E}|$	$|\mathcal{R}|$	#Train	#Valid	#Test
ogbl-wikikg2	2,500k	535	16,109k	429k	598k
ogbl-biokg	94k	51	4763k	163k	163k
YAGO3-10	123k	37	1,079k	5k	5k
DB100K	100k	470	597k	50k	50k
FB15K	15k	237	483k	50k	50k
WN18	41k	18	141k	5k	5k
FB15K-237	15k	237	272k	18k	20k
WN18RR	41k	11	87k	3k	3k
Sports	1039	4	1312	-	307
Location	445	5	384	-	100

Table 1: Statistics of the datasets in the experiment.

5. Experimental Setup

5.1. Datasets

We utilize the 10 most commonly utilized link prediction datasets and validate the effectiveness and generalizability of TransERR. We summarise the details of datasets in Table 1. ogbl-wikikg2 (Hu et al., 2020) is a very large-scale dataset extracted from the Wikidata knowledge base (Vrandečić and Krötzsch, 2014). ogbl-biokg (Hu et al.,

2020) consists massive biomedical data. YAGO3-10 (Suchanek et al., 2007) is a subset of YAGO3, which are mainly from Wikipedia. DB100k (Ding et al., 2018) is a subset of DBpedia. FB15K (Bordes et al., 2013) is a subset of the Freebase knowledge base, while FB15k-237 (Toutanova and Chen, 2015) removes the inversion relations from FB15K. WN18 (Bordes et al., 2013) is extracted from WordNet (Miller, 1995). WN18RR (Dettmers et al., 2018b) removes inversion relations similar to FB15K-237. Sports (Wang et al., 2015) and Location (Wang et al., 2015) are both small-scale datasets. They are the main subrelation pattern. They are the NELL's (Mitchell et al., 2018) subsets.

5.2. Evaluation Protocol

Following the SOTA methods (Sun et al., 2019; Chao et al., 2021), the quality of the ranking of each test triplet is evaluated via calculating all possible substitutions of head entity and tail entity : (h', r, t) and (h, r, t'), where $h', t' \in \mathcal{E}$. We evaluate the performance using the standard evaluation metrics, including Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hits@N. Hits@N measures the percentage of correct entities in the top n predictions. Higher values of MRR and Hits@N indicate better performance. Nevertheless, MR is the exact opposite of MRR and Hits@n. Hits@N ratio with cut-off values $N = 1, 3, 10$. For experiments on ogbl-wikikg2 and ogbl-biokg, we follow the evaluation protocol of these two benchmarks (Hu et al., 2020).

5.3. Implementation

We implement our proposed model via pytorch. We use Adam (Kingma and Ba, 2014) optimizer, and employ grid search to find the best hyperparameters based on the performance on the validation datasets. We report averaged test results across ten runs and use the random seeds from 0 to 9. We omit the variance as it is generally low. We employ the official implementations (Hu et al., 2020) for ogbl-wikikg2 [1]. In general, the embedding size d is tuned amongst $\{100, 200, 500, 1000, 1500, 2000\}$, the learning rate ϵ is tuned amongst $\{1e^{-3}, 5e^{-4}, 1e^{-4}, 5e^{-5}, 1e^{-5}\}$, the negative sample n is selected in $\{128, 256\}$, the self-adversarial sampling temperature α is selected from $\{0.5, 1\}$, the fixed margin γ are searched from 5 to 30. All optimal hyperparameters are selected on the validation set.

---

[1] https://ogb.stanford.edu/

	ogbl-wikikg2				ogbl-biokg			
	#Dim	#Params	Test MRR	Valid MRR	#Dim	#Params	Test MRR	Valid MRR
TransE	500	1,250M	0.4256	0.4272	2,000	187M	0.7452	0.7456
DistMult	500	1,250M	0.3729	0.3506	2,000	187M	0.8043	0.8055
ComplEx	250	1,250M	0.4027	0.3759	1,000	187M	0.8095	0.8105
RotatE	250	1,250M	0.4332	0.4353	1,000	187M	0.7989	0.7997
Rot_Pro	200	1,000M	0.5602	0.5740	-	-	-	-
PairRE	200	500M	0.5208	0.5423	2,000	187M	0.8164	0.8172
TripleRE	200	500M	0.5794	0.6045	2,000	187M	0.8191	0.8192
TranSHER	200	500M	0.5536	0.5662	2,000	187M	<u>0.8233</u>	<u>0.8244</u>
TransERR	100	**250M**	<u>0.6100</u>	<u>0.6246</u>	1,000	**93M**	0.8153	0.8156
TransERR	200	500M	**0.6359**	**0.6518**	2,000	187M	**0.8243**	**0.8249**

Table 2: Results on ogbl-wikikg2 and ogbl-biokg. Results are taken from the official leaderboard (Hu et al., 2020). The dashes mean that the results are not reported in the responding literature.

	WN18RR					FB15K-237				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE ♡	3384	0.266	0.501	-	-	357	0.294	0.465	-	-
DistMult ♡	5110	0.43	0.49	0.44	0.39	254	0.241	0.419	0.263	0.155
ComplEx ♡	5261	0.44	0.51	0.46	0.41	339	0.247	0.428	0.275	0.158
TuckER	-	0.470	0.526	0.482	0.443	-	0.358	0.544	0.394	**0.266**
RotatE ♡	3340	0.476	0.571	0.492	0.428	177	0.338	0.533	0.375	0.241
Rotat3D	3328	0.489	0.579	0.505	0.442	165	0.347	0.543	0.385	0.250
QuatE	3472	0.481	0.564	0.500	0.436	176	0.311	0.495	0.342	0.221
Rot_Pro	2815	0.457	0.557	0.482	0.397	201	0.344	0.540	0.383	0.246
PairRE	-	-	-	-	-	160	0.351	0.544	0.387	0.256
TripleRE	-	-	-	-	-	142	0.351	0.544	0.387	0.256
TranSHER	-	-	-	-	-	-	**0.360**	0.551	**0.397**	0.264
TransERR	**1167**	**0.501**	**0.605**	**0.520**	**0.450**	**125**	**0.360**	**0.555**	0.396	0.264

Table 3: Results on WN18RR and FB15K-237. Results of ♡ are taken from Sun et al. (2019). The best results are in bold. The dashes mean that the results are not reported in the responding literature.

	WN18					FB15K				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE ♡	-	0.495	0.943	0.888	0.111	-	0.463	0.749	0.578	0.297
DistMult ♡	665	0.797	0.946	-	-	42	0.798	0.893	-	-
ComplEx ♡	-	0.941	0.947	0.945	0.936	-	0.692	0.840	0.759	0.599
TuckER	-	**0.953**	0.958	0.955	**0.949**	-	0.795	0.892	0.833	0.741
RotatE ♡	309	0.949	0.959	0.952	0.944	40	0.797	0.884	0.830	0.746
Rotat3D	214	0.951	0.961	0.953	0.945	39	0.789	0.887	0.835	0.728
QuatE	338	0.949	0.960	0.954	0.941	41	0.770	0.878	0.821	0.700
PairRE	401	0.941	0.956	0.950	0.940	37	0.811	**0.896**	0.845	0.765
TripleRE	-	-	-	-	-	**35**	0.747	0.877	0.813	0.662
TransERR	**82**	**0.953**	**0.965**	**0.957**	0.945	41	**0.815**	**0.896**	**0.848**	**0.767**

Table 4: Results on WN18 and FB15K. Results of ♡ are taken from Sun et al. (2019). Other results are taken from the corresponding original papers. The best results are in bold. The dashes mean that the results are not reported in the responding literature.

6. Results and Analysis

6.1. Main Results

To evaluate the effectiveness and the generalization of TransERR, we perform the experiments on 10 benchmark datasets of different scales. The results on ogbl-wikikg2 and ogbl-biokg are shown in Table 2. TransERR achieves competitive results on large-scale datasets. On ogbl-wikikg2, TransERR obtains significant improvements of 9.7% than TripleRE with the same number of parame-

	YAGO3-10					DB100K				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
TransE	-	-	-	-	-	-	0.111	0.270	0.164	0.016
DistMult	5926	0.34	0.54	0.38	0.24	-	0.233	0.448	0.301	0.115
ComplEx	6351	0.36	0.55	0.40	0.26	-	0.242	0.440	0.312	0.126
ConvE	1671	0.44	0.62	0.49	0.35	-	-	-	-	-
Rot_Pro	1797	0.542	0.699	0.596	0.443	867	0.359	0.599	0.471	0.306
PairRE	-	-	-	-	-	-	0.412	0.600	0.472	0.309
TranSHER	-	-	-	-	-	-	0.431	0.589	0.476	0.345
TransERR	**476**	**0.546**	**0.706**	**0.601**	**0.456**	571	**0.465**	**0.622**	**0.510**	**0.380**

Table 5: Results on YAGO3-10 and DB100K. Results are taken from the corresponding original papers. The best results are in bold. The dashes mean that the results are not reported in the responding literature.

	Sports					Location				
	MR	MRR	Hits@10	Hits@3	Hits@1	MR	MRR	Hits@10	Hits@3	Hits@1
SimplE ♠	-	0.230	0.324	0.234	0.184	-	0.190	0.315	0.210	0.130
SimplE$^+$ ♠	-	0.404	0.508	0.440	0.394	-	0.440	0.450	0.440	0.430
RotatE$^\diamond$	191	0.420	0.535	0.503	0.420	73	0.486	0.550	0.480	0.455
PairRE $^\heartsuit$	-	0.468	-	-	0.416	-	-	-	-	-
TranSHER$^\diamond$	151	0.479	0.537	0.509	0.433	71	0.475	0.575	0.470	0.435
TransERR	**95**	**0.499**	**0.570**	**0.526**	**0.447**	**30**	**0.563**	**0.645**	**0.565**	**0.520**

Table 6: Results on Sports and Location. Results of ♠ and ♡ are taken from Fatemi et al. (2019) and Chao et al. (2021), respectively. ◇ are obtained from our experiments. The best results are in bold. The dashes mean that the results are not reported in the responding literature.

ters (#Dim 200) in Test MRR. It is worth noting that TransERR still outperforms PairRE, TripleRE and TranSHER with fewer parameters (#Dim 100). In addition, TransERR also outperforms RotatE and translation-based models with fewer parameters on ogbl-biokg. The results demonstrate that TransERR has a powerful capability to model large-scale KGs.

The comparison results for WN18RR and FB15K-237 are shown in Table 3. TransERR outperforms all the baselines in all metrics on WN18RR. Compared with RotatE, TransERR achieves improvements of 65.0%, 5.2%, 5.9%, 5.6% and 5.1%, respectively. For FB15K-237, TransERR gains comparable results than existing distance-based models in MR, MRR and Hit@10. The above results confirm that TransERR has merit in modeling graph embeddings and improves the link prediction performance. This is because quaternions enable expressive rotation in the hypercomplex-valued space, and two unit quaternion vectors further narrow the translation distance between the head and tail entities.

The comparison results for WN18 and FB15K are shown in Table 4. We employ the same hyperparameter settings and implementation compared with the previous works. Since both the baselines and TransERR almost obtain the upper bound on Hits@N, the improvement of TransERR is still considered effective. We can see that TransERR achieves significant improvements on WN18 and FB15K.

The results on YAGO3-10 and DB100K datasets are shown in Table 5. TransERR achieves the best results in all metrics for YAGO3-10. On DB100K, compared with the latest TranSHER, TransERR produces the optimal performance in all metrics except MR, which obtains significant improvements of 7.8%, 5.6%, 7.1% and 10.1%, respectively.

We compare TransERR with the two latest methods SimplE$^+$ and PairRE, which are all focused on modeling subrelation pattern. As shown in Table 6, TransERR achieves the SOTA results on Sports and Location. Compared with the latest PairRE, TransERR obtains improvements of 6.6% and 7.4% in MRR and Hits@1 on Sports. For Location, TransERR outperforms all the baselines in all metrics.

6.2. Complex Relations Analysis

This section analyzes the performance of the proposed TransERR on complex relations patterns. Following Han et al. (2018), we classify the relations into four categories: 1-to-1, 1-to-N, N-to-1 and N-to-N. The results of TransERR on different relation categories on FB15K-237 are shown in Table 7. Compared with RotatE, TransERR obtains 2.5% and 7.3% significant improvements in MRR and Hits@10 for 1-1 scenario when predicting head en-

Relation Type	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
	Head prediction (MRR)				**Tail prediction (MRR)**			
TransE	0.494	0.456	0.083	0.252	0.481	0.073	0.751	0.365
DistMult	0.213	0.440	0.071	0.229	0.212	0.053	0.727	0.345
ComplEx	0.356	0.465	0.091	0.247	0.373	0.062	0.741	0.356
RotatE	0.502	0.465	0.092	0.259	0.486	0.078	0.756	0.375
PairRE	0.493	0.482	0.116	0.271	0.492	0.071	0.775	0.386
TripleRE	0.499	0.485	0.116	0.283	0.493	0.077	0.776	0.388
TranSHER	0.501	0.487	**0.119**	**0.285**	0.494	0.079	0.779	0.389
TransERR	**0.515**	**0.495**	0.117	0.284	**0.515**	**0.080**	**0.785**	**0.393**
	Head prediction (Hits@10)				**Tail prediction (Hits@10)**			
TransE	0.494	0.456	0.083	0.252	0.481	0.073	0.751	0.365
DistMult	0.452	0.640	0.139	0.421	0.449	0.115	0.839	0.559
ComplEx	0.452	0.643	0.142	0.423	0.445	0.114	0.845	0.563
RotatE	0.601	0.672	0.175	0.468	0.586	0.143	0.880	0.615
PairRE	0.603	0.670	0.213	0.482	0.599	0.149	0.892	0.620
TripleRE	0.611	0.671	0.215	0.486	0.601	0.154	0.890	0.620
TranSHER	0.615	0.674	0.211	0.488	0.604	**0.162**	0.891	0.624
TransERR	**0.645**	**0.696**	**0.226**	**0.495**	**0.619**	0.157	**0.897**	**0.632**

Table 7: Results on FB15k-237 by relation category. Best results are in bold. Head prediction: predicting h given $(?, r, t)$. Tail prediction: predicting t given $(h, r, ?)$.

tities. Besides, TransERR achieves 9.6% and 5.7% significant improvements in MRR and Hits@10 for N-N scenario. The results prove that TransERR possesses a powerful ability to capture the latent information on complex relations (1-1, 1-N, N-1 and N-N) than the existing distance-based models.

6.3. Key Relation Patterns Analysis

To illustrate the learned relation patterns that include symmetry, antisymmetry, inversion and composition, we visualize the histogram of relations embeddings in some examples in Figure 2. The relations shown in Figure 2 are derived from FB15K.

Symmetry. In Figure 2a, $\mathbf{r_1}$ is a symmetry relation. According to the theoretical analysis mentioned in Section 4.2, TransERR can infer the symmetry relation pattern when $\mathbf{r_1^{\triangleleft H}} = -\mathbf{r_1^{\triangleleft T}}$ is satisfied. We can observe that $\mathbf{r_1^{\triangleleft H}} + \mathbf{r_1^{\triangleleft T}}$ tends to 0 as much as possible in Figure 2b.

Antisymmetry. On the contrary, $\mathbf{r_2}$ is an antisymmetric relation pattern in Figure 2c. TransERR encodes antisymmetry relation pattern when $\mathbf{r_2^{\triangleleft H}} \neq -\mathbf{r_2^{\triangleleft T}}$ is satisfied. We can find that Figure 2d satisfies the above condition.

Inversion. $\mathbf{r_2}$ and $\mathbf{r_3}$ are an example of inversion relation pairs shown in Figure 2c and Figure 2e, respectively. TransERR infers the inversion relation pattern when $\mathbf{r_2^{\triangleleft H}} \otimes \mathbf{r_3^{\triangleleft H}} = \mathbf{r_2^{\triangleleft T}} \otimes \mathbf{r_3^{\triangleleft T}}$ and $\mathbf{r_2} = -\mathbf{r_3}$ are satisfied. Figure 2f and Figure 2g show that TransERR satisfies the above conditions.

Composition. $\mathbf{r_4}$, $\mathbf{r_5}$, and $\mathbf{r_6}$ are an example of composition relation pattern shown in Figure 2h, Figure 2i and Figure 2j, respectively. TransERR infers the composition relation pattern when $\mathbf{r_6^{\triangleleft H}} =$

$\mathbf{r_4^{\triangleleft H}} \otimes \mathbf{r_5^{\triangleleft H}}$, $\mathbf{r_6^{\triangleleft T}} = \mathbf{r_4^{\triangleleft T}} \otimes \mathbf{r_5^{\triangleleft T}}$ and $\mathbf{r_6} = \mathbf{r_4} \otimes \mathbf{r_5^{\triangleleft H}} + \mathbf{r_5} \otimes \mathbf{r_4^{\triangleleft T}}$ are satisfied. We can observe that Figure 2k, Figure 2l and Figure 2m satisfy the above conditions, respectively.

Subrelation. To verify that TransERR can infer the subrelation pattern, we employ the hard rule constraint for subrelation. For (e_1, r_1, e_2) and (e_1, r_2, e_2), we add the rules $(r1 \rightarrow r2)$ are defined as follows:

$$\mathbf{r_2^{\triangleleft H}} = \mathbf{r_1^{\triangleleft H}}\mu, \quad \mathbf{r_2^{\triangleleft T}} = \mathbf{r_1^{\triangleleft T}}\mu, \quad \mathbf{r_2} = \mathbf{r_1}\mu. \quad (8)$$

We conduct experiments on Sports and add the following rules: *CoachesTeam → PersonBelongsToOrganization* and *AthleteLedSportsTeam → AthletePlaysForTeam*. The link prediction results are shown in Table 8. Compared with other models, which are focused on encoding subrelation pattern, TransERR achieves the SOTA results. Compared with PairRE+Rule, TransERR+Rule obtains improvements of 6.9% and 7.4% in MRR and Hits@1. Results further show that TransERR can effectively model subrelation relation pattern.

	Sports			
	MRR	Hits@10	Hits@3	Hits@1
SimplE$^+$	0.404	0.508	0.440	0.394
PairE+Rule	0.475	-	-	0.432
TransERR	0.499	0.570	0.526	0.447
TransERR+Rule	**0.508**	**0.581**	**0.527**	**0.464**

Table 8: Link prediction results on Sports. All results are taken from the corresponding original papers. The dashes mean that the results are not reported in the responding literature.
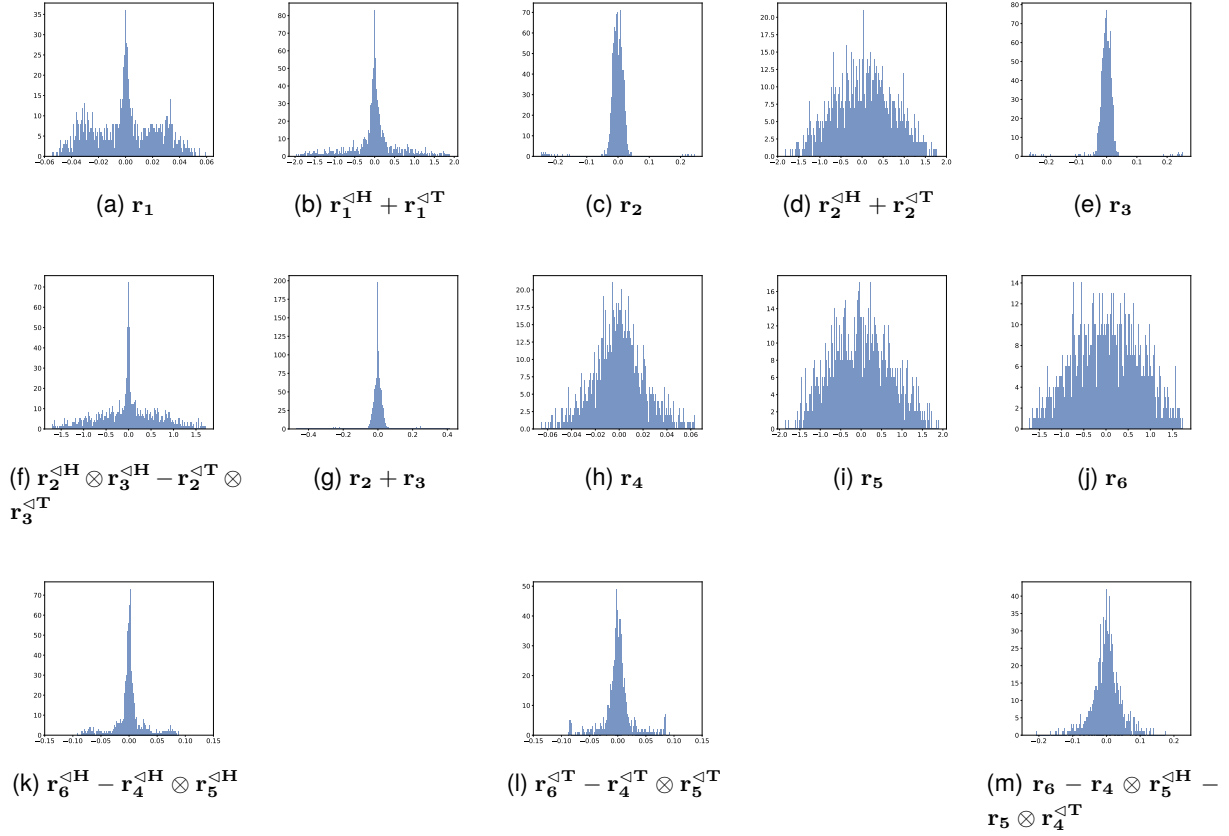
Figure 2: Relation embedding histograms for various relation patterns. $\mathbf{r_1}$ is */music/performance_role/regular_performances. /music/group_membership/role*. $\mathbf{r_2}$ and $\mathbf{r_3}$ are */film/actor/film./film/performance/film* and */film/film/starring./film/performance/actor*, respectively. $\mathbf{r_4}$, $\mathbf{r_5}$ and $\mathbf{r_6}$ are */people/person/nationality*, */location/location/contains* and */people/person/place_of_birth*, respectively.
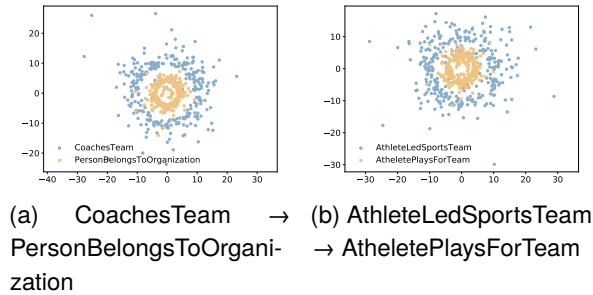
Figure 3: Visualisation of relation embeddings on Sports.

In addition, a quaternion number can be seen as a point on a 2D plane, and it contains a real part and three imaginary parts (i, j and k). We consider the three imaginary parts as a whole and plot the relation embeddings on a 2D plane. Moreover, we have $\frac{\mathbf{r_2^{\triangleleft H}}}{\mathbf{r_1^{\triangleleft H}}} = \frac{\mathbf{r_2^{\triangleleft T}}}{\mathbf{r_1^{\triangleleft T}}} = \frac{\mathbf{r_2}}{\mathbf{r_1}} = \mu$. In this study, the optimal embedding dimension of Sports is 200. Hence,

the dimension of $r^{\triangleleft H}$, r and $r^{\triangleleft T}$ are all 200. Since the scale factor μ is the same between them, we plot $[r^{\triangleleft H}, r, r^{\triangleleft T}]$ and the embedding dimension is 600. That is, each relation contains 600 points. Furthermore, we employ the logarithmic scale to better display the differences between relation embeddings. The results are shown in Figure 3. We observe that TransERR can learn the hierarchy between relations and effectively model subrelation pattern with the hard rule constraint. In general, the above experimental analysis can once again prove the ability of our model in modeling key relation patterns.

6.4. Ablation Study

In this part, in order to demonstrate that TransERR can better capture latent information between entity embeddings in the hypercomplex-valued space than in the complex-valued space, we encode TransERR in the complex space and conduct experiments on FB15K, FB15K-237, WN18 and

	FB15K		FB15K-237		WN18		WN18RR	
Scoring Function	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10
$-\parallel \mathbf{h} \circ \mathbf{r}^{\mathbf{H}} + \mathbf{r} - \mathbf{t} \circ \mathbf{r}^{\mathbf{T}} \parallel$	0.391	0.561	0.235	0.419	0.909	0.944	0.469	0.563
$-\parallel \mathbf{h} \circ \mathbf{r}^{\blacktriangleleft \mathbf{H}} + \mathbf{r} - \mathbf{t} \circ \mathbf{r}^{\blacktriangleleft \mathbf{T}} \parallel$	0.732	0.860	0.352	0.537	0.921	0.963	0.481	0.580
$-\parallel \mathbf{h} \otimes \mathbf{r}^{\mathbf{H}} + \mathbf{r} - \mathbf{t} \otimes \mathbf{r}^{\mathbf{T}} \parallel$	0.425	0.598	0.290	0.454	0.950	0.961	0.492	0.586
$-\parallel \mathbf{h} \otimes \mathbf{r}^{\triangleleft \mathbf{H}} + \mathbf{r} - \mathbf{t} \otimes \mathbf{r}^{\triangleleft \mathbf{T}} \parallel$	**0.815**	**0.896**	**0.360**	**0.555**	**0.953**	**0.965**	**0.501**	**0.605**

Table 9: Ablation of TransERR on FB15K, FB15K-237, WN18 and WN18RR. $\circ$ and $\otimes$ are Hadamard product and Hamilton product, respectively. $\mathbf{r}^{\blacktriangleleft \mathbf{H}}$ and $\mathbf{r}^{\blacktriangleleft \mathbf{T}}$ are normalized complex vectors. $\mathbf{r}^{\triangleleft \mathbf{H}}$ and $\mathbf{r}^{\triangleleft \mathbf{T}}$ are normalized quaternion vectors.

WN18RR. As shown in the second and fourth rows of Table 9, we can observe that TransERR behaves better in the quaternion space. This is further evidence that TransERR can facilitate interaction information between the head and tail entities in the hypercomplex-valued space. Furthermore, we conduct an ablation study on quaternion normalization for TransERR, where rows one and two are a control group and rows three and four are a control group in Table 9. We remove the normalization step for $\mathbf{r}^{\mathbf{H}}$ and $\mathbf{r}^{\mathbf{T}}$ and utilize $\mathbf{r}^{\mathbf{H}}$ and $\mathbf{r}^{\mathbf{T}}$ to rotate head entities and tail entities in the complex-valued space and in the hypercomplex-valued space, respectively. We conclude that the relational rotation's geometric property is lost, leading to poor performance. In addition, the unit quaternion has a higher degree of smooth rotation and spatial transformation ability than the unit complex number.

7. Conclusion

This paper proposes a simple yet effective distance-based KGE model (TransERR), which rotates entities with two normalized quaternion vectors in the hypercomplex-valued space. TransERR possesses a higher degree of translation freedom for graph embeddings. We provide formal mathematical proofs to demonstrate that TransERR can encode the key relation patterns. Moreover, the results show that TransERR can effectively model complex relation patterns, including 1-1, 1-N, N-1 and N-N. The experiments also suggest that TransERR can maximize interaction information between entities in the hypercomplex-valued space. The experimental results fully illustrate the effectiveness and generalizability of our model. In addition, the results show that two unit quaternions can further narrow the distance between the head and tail entities, and thus avoid information loss in rotation.

8. Acknowledgement

9. Bibliographical References

Ivana Balažević, Carl Allen, and Timothy Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 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 *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. Pairre: Knowledge graph embed-

dings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4360–4369.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Bidirectional attentive memory networks for question answering over knowledge bases. *arXiv preprint arXiv:1903.02188*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018a. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018b. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 110–121.

Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. 2019. Improved knowledge graph embedding using background taxonomic information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3526–3533.

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 *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 385–394.

Wm R Hamilton. 1844. Theory of quaternions. *Proceedings of the Royal Irish Academy (1836-1869)*, 3:1–16.

Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.

Zikun Hu, Yixin Cao, Lifu Huang, and Tat-Seng Chua. 2021. How knowledge graph and attention help? a quantitative analysis into bag-level relation extraction. *arXiv preprint arXiv:2107.12064*.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In *Thirtieth AAAI conference on artificial intelligence*.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Ademar Crotti Junior, Fabrizio Orlandi, Damien Graux, Murhaf Hossari, Declan O'Sullivan, Christian Hartz, and Christian Dirschl. 2020. Knowledge graph-based legal search over german court cases. In *European Semantic Web Conference*, pages 293–297. Springer.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, 31.

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

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Jiang Li, Xiangdong Su, Xinlan Ma, and Guanglai Gao. 2022a. Quatse: Spherical linear interpolation of quaternion for knowledge graph embeddings. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 209–220. Springer.

Yizhi Li, Wei Fan, Chao Liu, Chenghua Lin, and Jiang Qian. 2022b. TranSHER: Translating knowledge graph embedding with hyperellipsoidal restriction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8517–8528, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation

embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, and Dinh Phung. 2022. Quatre: Relation-aware quaternions for knowledge graph embeddings. In *Companion Proceedings of the Web Conference 2022*, pages 189–192.

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 *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333.

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

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

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

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

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 *International conference on machine learning*, pages 2071–2080. PMLR.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, Nilesh Agrawal, and Partha Talukdar. 2020. Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3009–3016.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Twenty-fourth international joint conference on artificial intelligence*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Shihui Yang, Jidong Tian, Honglun Zhang, Junchi Yan, Hao He, and Yaohui Jin. 2019. Transms: Knowledge graph embedding for complex relations by multidirectional semantics. In *IJCAI*, pages 1935–1942.

Long Yu, Zhicong Luo, Huanyong Liu, Deng Lin, Hongzhu Li, and Yafeng Deng. 2022. Triplere: Knowledge graph embeddings via tripled relation vectors. *arXiv preprint arXiv:2209.08271*.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. *Advances in neural information processing systems*, 32.