

# Hyperbolic Geometry is Not Necessary: Lightweight Euclidean-Based Models for Low-Dimensional Knowledge Graph Embeddings

Kai Wang<sup>1,2,3\*</sup>, Yu Liu<sup>1,2</sup>, Dan Lin<sup>1,2</sup>, Quan Z. Sheng<sup>3</sup>

<sup>1</sup>School of Software Technology, Dalian University of Technology, Dalian, 116620, China

<sup>2</sup>Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, China

kai\_wang@mail.dlut.edu.cn, yuliu@dlut.edu.cn, lindan0823@mail.dlut.edu.cn

<sup>3</sup>Department of Computing, Macquarie University, Sydney, NSW 2109, Australia

michael.sheng@mq.edu.au

## Abstract

Recent knowledge graph embedding (KGE) models based on hyperbolic geometry have shown great potential in a low-dimensional embedding space. However, the necessity of hyperbolic space in KGE is still questionable, because the calculation based on hyperbolic geometry is much more complicated than Euclidean operations. In this paper, based on the state-of-the-art hyperbolic-based model RotH, we develop two lightweight Euclidean-based models, called *RotL* and *Rot2L*. The RotL model simplifies the hyperbolic operations while keeping the flexible normalization effect. Utilizing a novel two-layer stacked transformation and based on RotL, the Rot2L model obtains an improved representation capability, yet costs fewer parameters and calculations than RotH. The experiments on link prediction show that Rot2L achieves the state-of-the-art performance on two widely-used datasets in low-dimensional knowledge graph embeddings. Furthermore, RotL achieves similar performance as RotH but only requires half of the training time.

## 1 Introduction

To represent entities and relations of knowledge graphs (KGs) in the semantic vector space, researchers have proposed various knowledge graph embedding (KGE) models, which have shown great potential in knowledge graph completion and knowledge-driven applications (Wang et al., 2017; Broscheit et al., 2020). To achieve higher prediction accuracy, recent KGE models usually use high-dimensional embedding vectors up to 200 or even 500 dimensions (Sun et al., 2019; Zhang et al., 2019). However, when facing large-scale KGs with millions of entities, high embedding dimensions would require prohibitive training costs and storage space (Sachan, 2020; Xie et al., 2020). It hinders

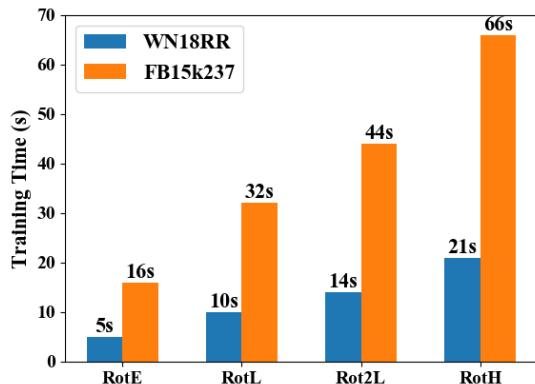


Figure 1: The training time per epoch of different 32-dimensional models on two datasets. All results are measured under the same parameter settings (500 batch size and 500 negative samples). RotL and Rot2L are the models proposed in this paper.

the practical application of KGE models, especially in mobile smart devices.

Recently, low-dimensional KGE models based on hyperbolic vector space have drawn some attention (Sun et al., 2020). The work of the first such model, MuRP, indicates that hyperbolic embeddings can capture hierarchical patterns in KGs and generate high-fidelity and parsimonious representations (Balazevic et al., 2019b). To capture logical patterns in KGs, Chami et al. propose a series of hyperbolic KGE models, including RotH, RefH, and AttH (Chami et al., 2020). Similar to the typical TransE model (Bordes et al., 2013) that treats the relation as a translation operation between the head and tail entity vectors, the state-of-the-art RotH model adjusts the head vector by the rotation and translation transformations to approach the tail vector in the hyperbolic space.

Although the above hyperbolic-based models outperform previous Euclidean-based models in low-dimensional condition, the necessity of hyperbolic space in this task is still questionable. Comparing a hyperbolic model with its Euclidean-based variant, it is uncertain which parts of the modifi-

\*The Corresponding Author

cation will be vital. Besides, despite theoretical support, the Möbius matrix-vector multiplication and Möbius addition operations in hyperbolic embeddings are far more complicated than Euclidean multiplication and addition. As shown in Fig. 1, RotH requires threefold more training time than its Euclidean-based variant RotE on two datasets. Especially on large-scale knowledge graphs, the additional calculating cost caused by the complicated hyperbolic operations would make the problem much severer.

Facing these problems, we analyze the effective components in the hyperbolic KGE models, and propose two lightweight “RotH-like” models, RotL and Rot2L, for low-dimensional knowledge graph embeddings. Without the hyperbolic geometry, RotL eliminates the Möbius matrix-vector multiplication and designs a new flexible addition operation to replace the Möbius addition. To further improve the RotL’s representation capability, the Rot2L model utilizes two stacked rotation-translation transformations in the Euclidean space. Benefiting from a specific parameterization strategy, Rot2L requires fewer parameters and calculations than RotH.

We conduct extensive experiments on two widely-used datasets. The results show that RotL outperforms existing Euclidean-based models in the 32-dimensional condition and only requires half of the training time of RotH. Rot2L obtains the state-of-the-art performance on the two datasets and outperforms RotH in both prediction accuracy and training speed. According to ablation experiments, we prove the effectiveness of the flexible addition and the other significant modules in Rot2L. We also verify our models in different embedding dimensions and analyze the performance difference between RotH and our models in a relation-specific experiment.

The rest of the paper is organized as follows. We discuss the background and definitions in Sec. 2. Sec. 3 introduces the technical details of RotL and Rot2L models. Sec. 4 reports the experimental studies and Sec. 5 further discusses several experimental investigations. The related work is reviewed in Sec. 6. Finally, we offer some concluding remarks in Sec. 7.

## 2 Background

In this section, we briefly describe the preliminaries related to this work.

### 2.1 Knowledge Graph Embeddings

In a knowledge graph  $\mathcal{G} = (E, R, T)$ ,  $E$  and  $R$  denote the set of entities and relations, and  $T$  is the collection of factual triples  $(h, r, t)$  where the head and tail entities  $h, t \in E$  and the relation  $r \in R$ .  $N_e$  and  $N_r$  refer to the number of entities and relations, respectively.

Knowledge graph embeddings aim to represent each entity  $e$  and each relation  $r$  as  $d$ -dimensional continuous vectors. A KGE model is evaluated by the link prediction task, which aims to find  $e_t \in E$  given an e-r query  $q = (e, r)$ , such that the triple  $(e, r, e_t)$  or  $(e_t, r, e)$  should belong to the knowledge graph  $\mathcal{G}$ . Generally, a scoring function  $F(h, r, t)$  is designed to measure each candidate triple. Take the distance-based scoring function  $F(h, r, t) = D(Q(h, r), t)$  as an example, it involves two operations: 1) *Transformation function*  $Q(h, r)$  transforms the head vector  $\mathbf{h}$  using the relation vector  $\mathbf{r}$ ; 2) *Distance function*  $D(\mathbf{q}, \mathbf{t})$  measures the distance between the tail vector  $\mathbf{t}$  and the transformed head vector  $\mathbf{q} = Q(h, r)$ .

### 2.2 Hyperbolic Geometry

Recently, researchers start to work on effective low-dimensional models in the KGE domain (Sachan, 2020; Wang et al., 2021a,b). Multiple hyperbolic KGE models, such as MuRP, RotH, RefH and AttH, have achieved good performance in low-dimensional condition (Balazevic et al., 2019b; Chami et al., 2020). These models employ a hyperbolic geometry model, the  $d$ -dimensional Poincaré ball (Birman and Ungar, 2001) with negative curvature  $-c$  ( $c > 0$ ):  $\mathbb{B}_c^d = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|^2 < \frac{1}{c}\}$ , where  $\|\cdot\|$  denotes the  $L_2$  norm.

The hyperbolic space is one of the three kinds of isotropic spaces, and the relevant theoretical research has been carried out for decades (Birman and Ungar, 2001). To achieve the vector transformation in the hyperbolic space, the Möbius addition  $\oplus_c$  and Möbius matrix-vector multiplication  $\otimes_c$  are utilized. Möbius addition (Ungar, 2001) is proposed to approximate Euclidean addition in the hyperbolic space:

$$\mathbf{x} \oplus_c \mathbf{y} = \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2) \mathbf{x} + (1 - c\|\mathbf{x}\|^2) \mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2} \quad (1)$$

where  $\langle \cdot \rangle$  is the Euclidean inner product. It is clear that Möbius addition requires much more calculations than an ordinary addition.

Möbius matrix-vector multiplication (Ganea et al., 2018) is also more complicated than Eu-

clidean multiplication. Before computing matrix multiplication with  $M \in \mathbb{R}^{d \times k}$ , the vector  $x \in \mathbb{B}_c^d$  is projected onto the tangent space at  $\mathbf{0} \in \mathbb{B}_c^d$  with the *logarithmic map*  $\log_0^c(x)$ . Then the output of multiplication is projected back to  $\mathbb{B}_c^d$  via the *exponential map*  $\exp_0^c(x)$ , i.e.,

$$M \otimes_c \mathbf{x} = \exp_0^c(M \log_0^c(\mathbf{x})) \quad (2)$$

$$\exp_0^c(\mathbf{x}) = \tanh(\sqrt{c}\|\mathbf{x}\|) \frac{\mathbf{x}}{\sqrt{c}\|\mathbf{x}\|} \quad (3)$$

$$\log_0^c(\mathbf{x}) = \operatorname{arctanh}(\sqrt{c}\|\mathbf{x}\|) \frac{\mathbf{x}}{\sqrt{c}\|\mathbf{x}\|} \quad (4)$$

### 2.3 The RotH Model

We briefly review RotH (Chami et al., 2020), the state-of-the-art model in the low-dimensional KGE. According to the official PyTorch implementation<sup>1</sup>, the scoring function of RotH employs a “translation-rotation-translation” transformation and utilizes a hyperbolic distance as the distance function  $D$ .

Specifically, let  $e^H \in \mathbb{B}_c^d$  denote entity hyperbolic embeddings of entity  $e$ . For one relation  $r$ , two hyperbolic relation vectors  $r^H, r'^H \in \mathbb{B}_c^d$  are defined for two translation operations. Using a  $d$ -dimensional vector  $\hat{r}$ , RotH parameterizes a *Givens Rotation* operation with a block-diagonal matrix of the form:

$$\operatorname{Rot}(\hat{r}) = \operatorname{diag}(G(\hat{r}_1, \hat{r}_2), \dots, G(\hat{r}_{d-1}, \hat{r}_d)) \quad (5)$$

$$\text{where } G(\hat{r}_i, \hat{r}_j) := \begin{bmatrix} \hat{r}_i & -\hat{r}_j \\ \hat{r}_j & \hat{r}_i \end{bmatrix}. \quad (6)$$

Then, for a triple  $(h, r, t)$ , the scoring function  $F_H$  of RotH is defined as:

$$Q_H^c(h, r) = \operatorname{Rot}(\hat{r}) \otimes_c (h^H \oplus_c r^H) \oplus_c r'^H \quad (7)$$

$$D_H^c(\mathbf{q}, t) = -\frac{2}{\sqrt{c}} \operatorname{arctanh}(\sqrt{c}\|\mathbf{q} \oplus_c \mathbf{t}^H\|)^2 \quad (8)$$

$$F_H(h, r, t) = D_H^c(Q_H^c(h, r), t) + b_h + b_t, \quad (9)$$

where  $c_r > 0$  is the relation-specific curvature parameter, and  $b_e (e \in E)$  are entity biases which act as margins in the scoring function (Balazevic et al., 2019b; Chami et al., 2020).

The other hyperbolic models can be regarded as RotH’s variants using different relation transformations. In addition, RotE is a Euclidean-based RotH variant, and its scoring function is defined as:

$$F_E(h, r, t) = -\|(\operatorname{Rot}(\hat{r})\mathbf{h} + \mathbf{r}) - \mathbf{t}\|^2 + b_h + b_t, \quad (10)$$

where  $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ . Without complex hyperbolic calculations,  $F_E$  can be computed in linear time of the embedding dimensions.

## 3 The Methodology

The goal of this work is to design a high-efficiency low-dimensional KGE model by extracting the effective components in the RotH model and eliminating the redundancy.

We find that RotH performs noticeably well because of two reasons. The first reason is *rotation-translation transformation*. As proved in previous research (Sun et al., 2019; Chami et al., 2020), this specific transforming operation can infer different relation patterns in the KG. The second reason is *flexible normalization*. All entity vectors in the hyperbolic space satisfy  $\|\mathbf{e}\|^2 < \frac{1}{c}$  before and after transformation, while the curvature  $c$  is relation-specific and self-adaptive. As the representation capability of a low-dimensional vector space is limited, the effect of flexible normalization would be more obvious. It explains why RotH can outperform its Euclidean-based variant RotE in low-dimensional KGE tasks.

In this section, we first propose a lightweight model, called RotL, which remains the flexible normalization of RotH and simplifies the complex hyperbolic operations. The details of RotL will be described in Sec. 3.1. We further design the Rot2L model using two stacked rotation-translation transformations. Rot2L employs a novel parameterization strategy that can save half of parameters in the two-layer architecture, which is detailed in Sec. 3.2. The architectures of the four models mentioned above are illustrated in Fig. 2.

### 3.1 The RotL Model and Flexible Addition

The RotL model aims to achieve similar performance to RotH and minimize its computational complexity close to that of RotE. Comparing the scoring functions of RotH and RotE in Eq. 9 and 10, it is clear that the additional calculations of RotH are centered on Möbius addition and Möbius matrix-vector multiplication.

Therefore, we first eliminate the hyperbolic embeddings in RotL and initialize the entity vector  $e$  and the two relation vectors for rotation and translation in the Euclidean space, such that the relation transformation can be calculated using Euclidean addition and multiplication directly.

To achieve the flexible normalization, we propose Flexible Addition  $\oplus_\alpha$ , a simplified form of Möbius Addition, i.e.,

$$\mathbf{x} \oplus_\alpha \mathbf{y} = \frac{\alpha(\mathbf{x} + \mathbf{y})}{1 + \langle \mathbf{x}, \mathbf{y} \rangle}, \quad (11)$$

<sup>1</sup><https://github.com/HazyResearch/KGEmb>

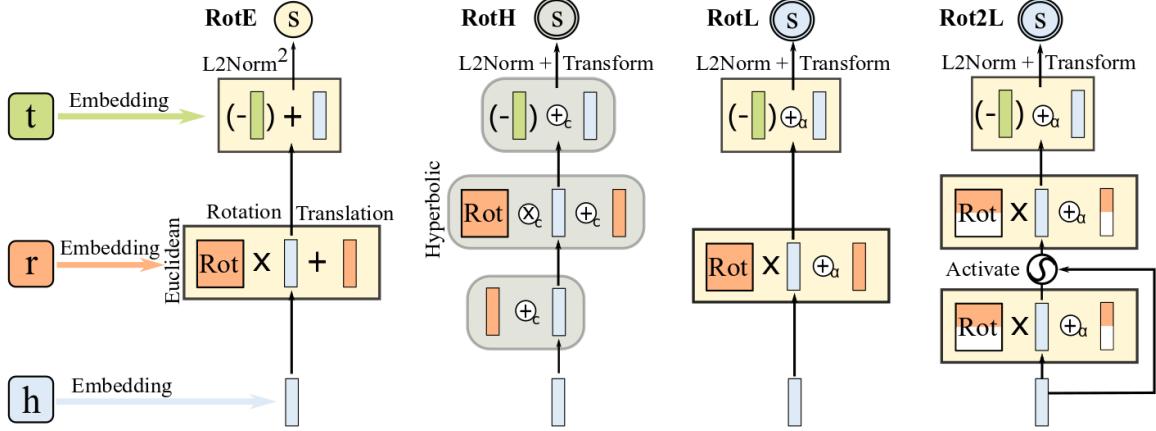


Figure 2: The architectures of four models, including the previous RotE and RotH, and the proposed RotL and Rot2L in this paper. The rectangle box denotes a Euclidean-based operation, while the rounded rectangle box denotes a hyperbolic-based one. The inside rectangles denote the embedding vectors or matrices, in which the relation-specific ones are in orange.  $\otimes_c$ ,  $\oplus_c$ ,  $\oplus_\alpha$  refer to Möbius multiplication, Möbius addition and Flexible addition, respectively.

where  $\alpha$  is a relation-specific scaling parameter and with a default value of 1. The Flexible Addition provides a self-adaptive normalization to  $(x + y)$ , and has lower computational complexity than Möbius addition. Counting the operation times of  $d$ -dimensional vector operations, the former requires three additions and two multiplications, while the latter needs nine and 12 operations, respectively. We further discuss the connection between the two operations through Theorem 1.

**Theorem 1.** Given that  $c = \alpha = 1$ , the Möbius Addition  $\oplus_c$  and Flexible Addition  $\oplus_\alpha$  satisfy that  $\mathbf{x} \oplus_c \mathbf{x}' \equiv \mathbf{x} \oplus_\alpha \mathbf{x}'$ .

*Proof.* With  $c = \alpha = 1$  and two vectors  $\mathbf{x}, \mathbf{x}' \in \mathcal{R}^d$ , satisfying  $\mathbf{x} = \mathbf{x}'$ ,

$$\begin{aligned} \mathbf{x} \oplus_c \mathbf{x}' &= \frac{(1 + 2\langle \mathbf{x}, \mathbf{x}' \rangle + \|\mathbf{x}'\|^2) \mathbf{x} + (1 - \|\mathbf{x}\|^2) \mathbf{x}'}{1 + 2\langle \mathbf{x}, \mathbf{x}' \rangle + \|\mathbf{x}\|^2 \|\mathbf{x}'\|^2} \\ &= \frac{(1 + \langle \mathbf{x}, \mathbf{x}' \rangle + \langle \mathbf{x}', \mathbf{x} \rangle + \|\mathbf{x}'\|^2) \mathbf{x} + (1 - \|\mathbf{x}\|^2) \mathbf{x}'}{1 + \langle \mathbf{x}, \mathbf{x}' \rangle + \langle \mathbf{x}', \mathbf{x} \rangle + \|\mathbf{x}\|^2 \|\mathbf{x}'\|^2} \\ &= \frac{(1 + \langle \mathbf{x}', \mathbf{x} \rangle)(\mathbf{x} + \mathbf{x}')}{(1 + \langle \mathbf{x}, \mathbf{x}' \rangle)(1 + \langle \mathbf{x}', \mathbf{x} \rangle)} = \frac{\mathbf{x} + \mathbf{x}'}{1 + \langle \mathbf{x}, \mathbf{x}' \rangle} = \mathbf{x} \oplus_\alpha \mathbf{x}' \end{aligned} \quad (12)$$

We emphasize that Theorem 1 indicates the equivalence of the two operations in a special condition. In our models, the proposed Flexible Addition is not equal to the Möbius Addition. It imitates the flexible normalization of the latter and eliminates the Hyperbolic space assumption.

We then define the transformation function of RotL as  $Q_L^\alpha(h, r) = \text{Rot}(\hat{r})\mathbf{h} \oplus_\alpha \mathbf{r}'$ , which can be regarded as a RotE transformation using the flexi-

ble addition. To fit this novel operation, we further modify the distance function of RotH in Eq. 8 by designing a simpler non-linear mapping operation. The distance function and scoring function of RotL are defined as follows:

$$D_L^\alpha(\mathbf{q}, t) = -\varphi(\|\mathbf{q} \oplus_\alpha \mathbf{t}\|) \quad (13)$$

$$F_L(h, r, t) = D_L^{\alpha'_r}(Q_L^{\alpha_r}(h, r), t) + b_h + b_t, \quad (14)$$

where  $\alpha_r$  and  $\alpha'_r$  are two different scaling parameters, and  $\varphi(x) = xe^x$  is empirically discovered to replace the arctanh function in RotH with less complexity.

Comparing Eq. 9 and 14, it is clear that the hyperbolic calculations are completely eliminated in the RotL model. Thus, RotL can reduce the computation complexity of RotH and save half of the training time as shown in Fig. 1.

### 3.2 The Rot2L Model and Stacked Transformation

Although the lightweight RotL maintains the flexible normalization effect, its performance is limited by the original transformation function of RotH. In this section, we describe a novel Rot2L model utilizing two stacked translation-rotation transformations.

According to the theory of affine transformation (Berger, 1987), the two transformations can be replaced by a single one. Therefore, inspired by neural networks, we design a two-layer architecture with an activate function in the middle, as shown

in Fig. 2. The transformation function  $Q_{2L}(h, r)$  in Rot2L is defined as:

$$\varrho(\mathbf{h}, \mathbf{q}) = \tanh(\mathbf{q}) + \gamma \mathbf{h} \quad (15)$$

$$Q_{2L}^{\alpha 1, \alpha 2}(h, r) = Q_L^{\alpha 1}(\varrho(\mathbf{h}, Q_L^{\alpha 2}(h, r)), r), \quad (16)$$

where  $\gamma$  is a hyper-parameter that balances the two parts.  $Q_L^{\alpha 1}$  and  $Q_L^{\alpha 2}$  represent two transformation layers, which are the same as the transformation function in RotL.

In the Rot2L model, the two layers need different parameters. This would double the amount of relation parameters because each layer requires two  $N_r \times d$  embedding matrices to represent the translation vectors and rotation matrices for all relations. To reduce relation parameters, Rot2L employs a novel parameterization strategy, which shares partial parameters among different relations.

Specifically, we utilize one embedding matrix  $\mathbf{M} \in \mathbb{R}^{N_r \times d}$  and a  $d$ -dimensional learnable vector  $\mathbf{f} \in \mathbb{R}^d$  for each rotation-translation transformation layer. Such that half of the parameters are shared in different relations by replacing another embedding matrix to the vector  $\mathbf{f}$ . Given the vector  $\mathbf{r} = \mathbf{M}[r]$  for the relation  $r$ , the corresponding translation vector and rotation matrix are constructed as:

$$\mathbf{r}' = [r_1, f_1, r_2, f_2, \dots, r_{\frac{d}{2}}, f_{\frac{d}{2}}], \quad (17)$$

$$Rot(\hat{r}) = diag(G(r_{\frac{d}{2}+1}, f_{\frac{d}{2}+1}), \dots, G(r_d, f_d)). \quad (18)$$

Finally, the scoring function of Rot2L contains the transformation function  $Q_{2L}(h, r)$  and the same distance function as RotL, which is defined as:

$$F_L(h, r, t) = D_L^{\alpha'_r}(Q_{2L}^{\alpha 1_r, \alpha 2_r}(h, r), t) + b_h + b_t. \quad (19)$$

Note that, it might be feasible to employ more transformation layers in Rot2L like deep neural networks. There are two reasons that we do not utilize more than two layers. First, more layers require more parameters, which goes against our original intention of being lightweight. Second, we find the vector values are gradually magnified when getting through multiple layers. Using three layers in the Rot2L model already suffers performance decrease. Exploring a deeper model with more effective regularization will be our future work.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** Our experimental studies are conducted on two widely-used datasets. WN18RR (Bordes et al., 2014) is a subset of the English lexical database WordNet (Miller, 1992), while FB15k237 (Toutanova and Chen, 2015) is extracted from Freebase including knowledge facts on movies, actors, awards, and sports. Inverse relations are removed from the two datasets, as many test triples can be obtained simply by inverting triples in the training set. The statistics of the datasets are given in Table 1 and “Train”, “Valid”, “Test” refer to the amount of triples in training, validation, and test sets.

Table 1: Statistics of the datasets.

Dataset	$N_r$	$N_e$	#Train	#Valid	#Test
FB15k237	237	14,541	272,115	17,535	20,466
WN18RR	11	40,943	86,845	3,034	3,134

**Implementation Details.** Following the previous work, we utilize a binary cross-entropy loss, which is defined as:

$$L = -\log\sigma(F(h, r, t)) - \sum_{i=0}^k \log(1 - \sigma(F(h'_i, r, t'_i))), \quad (20)$$

where  $\sigma(\cdot)$  refers to the Sigmoid function, and  $(h'_i, r, t'_i)$  refers to the negative samples after deleting training triples. All experiments are performed on NVIDIA GeForce GTX1080Ti GPUs, and implemented in Python using the PyTorch framework.

**Hyperparameter Settings.** According to the low dimensional condition, we train our model setting the embedding dimensions in  $\{8, 16, 32, 64\}$ , with the Adam optimizer for the WN18RR dataset and Adagrad optimizer for FB15k237. We select the hyper-parameters of our model via grid search according to the metrics on the validation set. Specifically, we empirically select the learning rate among  $\{0.0005, 0.005, 0.05\}$ , the batch size among  $\{100, 200, 500\}$ , the amount of negative samples  $k$  among  $\{50, 200, 500\}$ , and the balance hyper-parameter  $\gamma$  among  $\{0.1, 0.3, 0.5, 1.0\}$ .

**Evaluation Metrics.** For the link prediction experiments, we adopt three evaluation metrics: 1) MRR, the average inverse rank of the test triples, 2) Hits@10, the proportion of correct entities ranked in top 10, and 3) Hits@1, the proportion of correct entities ranked first. Higher MRR, Hits@10, and Hits@1 mean better performance. Following the

Table 2: The link prediction results on the WN18RR and FB15k237 datasets. The best scores of 32-dimensional models are in **Bold**.

Type	Methods	FB15K237			WN18RR		
		MRR	Hits@10	Hits@1	MRR	Hits@10	Hits@1
Euclidean-based Models	RotatE	0.290	0.458	0.208	0.387	0.491	0.330
	TuckER	0.306	0.475	0.223	0.428	0.474	0.401
	MuRE	0.313	0.489	0.226	0.458	0.525	0.421
	RefE	0.302	0.474	0.216	0.455	0.521	0.419
	RotE	0.307	0.482	0.220	0.463	0.529	0.426
	AttE	0.311	0.488	0.223	0.456	0.526	0.419
Hyperbolic-based Models	MuRP	0.323	0.501	0.235	0.465	0.544	0.420
	RefH	0.312	0.489	0.224	0.447	0.518	0.408
	RotH	0.314	0.497	0.223	0.472	0.553	0.428
	AttH	0.324	0.501	0.236	0.466	0.551	0.419
Our Models	<b>RotL</b>	0.320	0.500	0.229	0.469	0.550	0.426
	<b>Rot2L</b>	<b>0.326</b>	<b>0.503</b>	<b>0.237</b>	<b>0.475</b>	<b>0.554</b>	<b>0.434</b>

previous works, we process the output sequence in the filter mode.

## 4.2 Link Prediction Task

We evaluate our models in the link prediction task on the two datasets. The experimental results are shown in Table 2. We select ten compared models in two types, in which the first six models are Euclidean-based, and the others utilize hyperbolic embeddings. Following the setting of Chami et al. (Chami et al., 2020), all these models are in 32-dimensional vector space.

From the results, we have the following observations. At first, the four hyperbolic-based models generally outperform their Euclidean variants and RotatE and TuckER, which are the state-of-the-art models for high-dimensional knowledge graph embeddings. This proves the effectiveness of hyperbolic models in low-dimensional knowledge graph embeddings.

RotL outperforms RotE and the other Euclidean-based models. Compared with RotE, the Hits@10 of RotL improves from 0.529 to 0.550 on WN18RR, and from 0.482 to 0.500 on FB15k237. Using a lightweight architecture, RotL even achieves similar prediction accuracy as RotH on FB15k237. It indicates that the hyperbolic embeddings technology is possible to be replaced with flexible addition and new distance function. Using a novel two-layer transformation function, Rot2L further improves RotL and achieves the state-of-the-art results on two datasets. Especially compared with RotH, MRR of Rot2L improves from 0.314 to 0.326 on FB15k237, and the Hits@1 increases from 0.428 to 0.434 on WN18RR.

It should be noted that improving low-dimensional performance is much harder than that

in the high-dimensional condition. Our experimental results prove the effectiveness of Rot2L, while the computational complexity of Rot2L is lower than RotH and AttH.

## 4.3 Ablation Studies

We further conduct a series of ablation experiments to evaluate the different modules of our models. Two main improvements should be evaluated: 1) the new distance function (Dis) in Eq. 13 and 2) the middle activate function (Mid) in Eq. 15. Accordingly, we test the variants by eliminating one of the two functions (e.g., Rot2Lw/oDis by removing the distance function). The other parts, such as flexible addition and stacked transformations, can be verified by comparing RotE (a Euclidean-based variant of RotH), RotL and Rot2L. The experimental results are shown in Table 4.

From the results, we can see that Hits@10 of Rot2L are higher than Rot2Lw/oDis on the both datasets, which proves the effectiveness of the distance function. Similar result is also shown in RotL, but the improvement is relatively small. Hits@10 of Rot2Lw/oMid are lower than that of Rot2L on FB15k237, while having no obvious difference on WN18RR. This indicates that the activate function is more effective on FB15k237, which contains much more relations than WN18RR. For FB15k237, RotL outperforms Rot2Lw/oMid, indicating that facing complex relationships, the pure two-layer transformation is no better than a single layer. This further validates the contribution of the activate function. Comparing RotLw/oDis and RotE, the impact of flexible addition is obvious. Using a simple scaling operation, the flexible addition provides a 1% and 2% improvements of Hits@10 on the two datasets, respectively.

Table 3: Comparison of Hits@10 on relation-specific triples in WN18RR.  $\Delta(L-H)$  denotes the growth rate of RotL relative to RotH, and similarly  $\Delta(2L-H)$  denotes the growth rate of Rot2L. Higher  $K_{hsG}$  and lower  $\xi_G$  mean more hierarchical.

Relation	$K_{hsG}$	$\xi_G$	RotE	RotH	RotL	$\Delta(L-H)$	Rot2L	$\Delta(2L-H)$
hypernym	1.00	-2.46	0.242	0.250	0.247	-1.12%	<b>0.254</b>	1.44%
derivationally related form	0.07	-3.84	0.960	<b>0.970</b>	0.967	-0.24%	0.968	-0.15%
instance hypernym	1.00	-0.82	<b>0.533</b>	0.480	0.496	3.42%	0.467	-2.57%
verb group	0.07	-0.50	<b>0.974</b>	<b>0.974</b>	<b>0.974</b>	0.00%	<b>0.974</b>	0.00%
also see	0.36	-2.09	0.634	0.643	0.625	-2.78%	<b>0.688</b>	6.95%
has part	1.00	-1.43	0.343	<b>0.349</b>	0.329	-5.83%	0.334	-4.17%
<b>member of domain usage</b>	1.00	-0.74	0.375	0.292	<b>0.417</b>	42.86%	<b>0.417</b>	42.86%
<b>member of domain region</b>	1.00	-0.78	0.365	0.269	0.385	42.86%	<b>0.442</b>	64.27%
similar to	0.07	-1.00	<b>1.000</b>	<b>1.000</b>	0.667	-33.34%	<b>1.000</b>	0.00%
member meronym	1.00	-2.90	0.342	0.395	0.405	2.49%	<b>0.451</b>	14.00%
synset domain topic of	0.99	-0.69	0.382	0.408	0.439	7.53%	<b>0.443</b>	8.61%

Table 4: The results of ablation experiments on (a) FB15k237 and (b) WN18RR.

Methods	FB15K237		
	MRR	Hits@10	Hits@1
Rot2L	<b>0.326</b>	<b>0.503</b>	<b>0.237</b>
Rot2Lw/oMid	0.317	0.497	0.229
Rot2Lw/oDis	0.316	0.492	0.226
RotL	0.320	0.500	0.229
RotLw/oDis	0.316	0.495	0.227
RotE	0.307	0.482	0.220

(a)

Methods	WN18RR		
	MRR	Hits@10	Hits@1
Rot2L	<b>0.475</b>	<b>0.554</b>	<b>0.434</b>
Rot2Lw/oMid	0.474	0.554	0.435
Rot2Lw/oDis	0.463	0.543	0.426
RotL	0.469	0.550	0.426
RotLw/oDis	0.467	0.549	0.420
RotE	0.463	0.529	0.426

(b)

Overall, the experimental results indicate the effectiveness of the major modules of our proposed models in this paper. Based on the same Euclidean space, our RotL and Rot2L models have a significant performance improvement compared to the RotE model.

#### 4.4 Efficiency Analysis

We analyze and compare the computational complexity among RotE, RotH, RotL, and Rot2L in this section. In terms of time complexity, as shown in Fig. 1, RotL is much faster than RotH mainly because that the Flexible Addition requires only a quarter of the computational cost of the Möbius addition. Although Rot2L repeats the rotation-translation transformation twice, its computational cost is still lower than that of RotH.

In terms of space complexity, a slight difference is shown in the number of relation parameters. Because the parameters related to entities, including

entity embedding vectors and entity biases, are the same in the four models and occupy the vast majority of total parameters, RotH requires the most relation parameters,  $(3N_r + 1)d$ , including three relation transformation vectors and the learnable curvature for different relations. By contrast, RotE and RotL cost smaller, which are  $2N_r d$  and  $2(N_r + 1)d$ , and the extra part of RotL comes from  $\alpha$  in Flexible Addition. Although using an effective parameterization strategy, Rot2L still requires two shared vectors and another  $\alpha$ -related vector. Its relation parameter amount is  $(2N_r + 5)d$ . As the relation number  $N_r$  is always greater than four, the Rot2L model requires fewer parameters than RotH.

In summary, the RotL and Rot2L models are highly efficient and better than the RotH model in both time complexity and space complexity.

## 5 Discussion

In this section, we further discuss several important questions on the RotL and Rot2L models.

### Q1: Which parts of predictions are improved in our models comparing with RotH?

We measure the link prediction performance of relation-specific triples on WN18RR, shown in Table 3, to analyze the improvements of the Rot2L model. The results are generated in the 32-dimensional condition. Comparing in different relations, RotE and RotH have their own strengths. RotH has better Hits@10 in most relations but is weaker than RotE in the “member of domain usage” and “member of domain region” relations. RotL performs well like RotH, but fails to predict the “similar to” relation. As the optimal model, Rot2L obtains the best Hits@10 in 8 out of 11 relations and only has a small decrease in the other

three relations. It should be noted that RotL and Rot2L effectively improve on the two “member of” relations, comparing to RotH. Especially, Rot2L achieves 42.86% and 64.27% improvements than RotH on these relations. Achieving the flexible normalization of RotH in the Euclidean space, our models perform well in both RotH-dominant and RotE-dominant relations.

### Q2: Can our models encode hierarchical patterns like hyperbolic-based models?

As RotH has been proved on the benefits of hyperbolic geometric on hierarchical relations, we further analyze whether our models can still preserve this property. Following the work of Chami et al. (Chami et al., 2020), we utilize the Krackhardt hierarchy score ( $Khs_G$ ) and estimated curvature ( $\xi_G$ ) as metrics. The related results can be found in Table 4, in which a relation with higher  $Khs_G$  and lower  $\xi_G$  is more hierarchical.

In terms of non-hierarchical relations, such as “verb group”, Euclidean-based and hyperbolic-based RotH have similar performances. In terms of hierarchical relations satisfying  $Khs_G = 1$ , we observe that hyperbolic embeddings work better on relations having low  $\xi_G$ , such as “hypernym”, “has part”, and “member meronym”. Meanwhile, RotE and RotL outperform RotH in relations having relative higher  $\xi_G$ , such as “instance hypernym”, “member of domain usage”, and “member of domain region”. Compared with the other three models, Rot2L obtains the best Hits@10 in most relations and works effectively on hierarchical relations with different  $\xi_G$ .

The results indicate that the simplified models, RotL and Rot2L, have a good ability to encode hierarchical relations. They preserve the good properties of both hyperbolic geometric and the Euclidean-based RotE.

### Q3: How about the model performance in other embedding dimensions?

We further compare the four models in different dimensions from 8 to 128. The experimental results are shown in Fig. 3(a). The prediction accuracy of the four models improves with the growth of the embedding dimensions. When the dimensions are lower than 32, RotE is obviously weaker than the others, but it performs well in the high-dimensional condition. Except for RotE, the other three models obtain similar results under high dimensions, but there are still some differences. Specifically, RotL performs better in lower dimensions and achieves

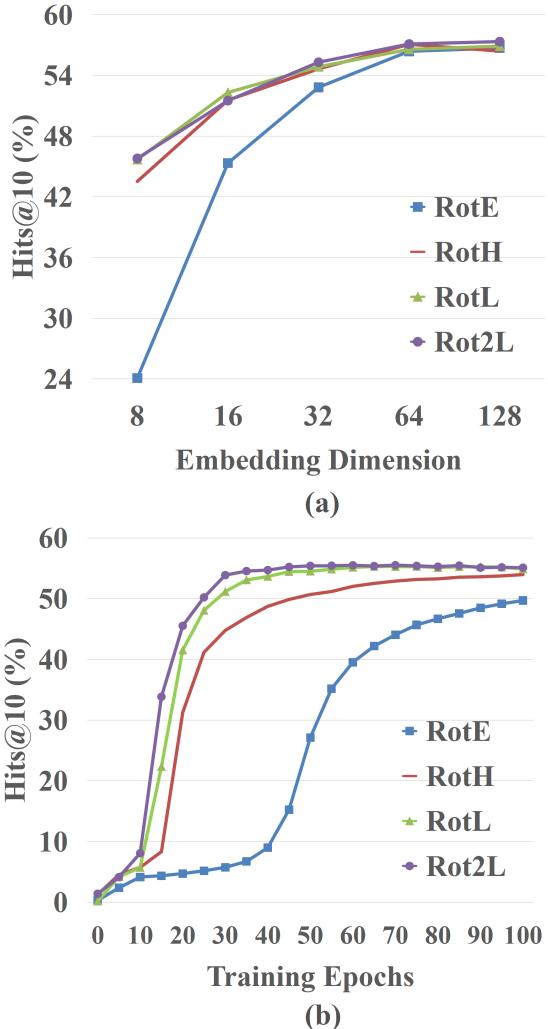


Figure 3: (a) The Hits@10 of four models on WN18RR with different embedding dimensions. (b) The changes of the Hits@10 on WN18RR as training proceeds.

the best Hits@10 under 16 dimensions. The advantage of Rot2L is shown with the increase of dimensions: Rot2L outperforms the other two after 32 dimensions. The experimental results prove that the stacked transformations in Rot2L have stronger representation capacity in the high-dimensional condition.

### Q4: Can our models accelerate the training speed?

Fig. 3 (b) shows the convergence of the training process for the four models under 32 dimensions. We can observe that RotE increases slowly in the first 40 epochs and converges later than the others. RotH converges faster than RotE, which is previously regarded as the contribution of hyperbolic space. From our experimental results, it is clear that both RotL and Rot2L show similar per-

formance. RotL, which has little difference from RotE in structure, shows a much faster training speed. Although RotE takes less time over one epoch, RotL can achieve higher performance with less training epochs. Comparing RotH and Rot2L, we can find that Rot2L precedes in almost every epoch. In the 25th epoch, Rot2L already achieves similar performance as the final performance of RotH. The results indicate that our models can replace the hyperbolic RotH model with comparable prediction accuracy and training speed.

## 6 Related Work

### 6.1 Knowledge Graph Embeddings

Various KGE models have been proposed using different scoring functions, such as translation-based TransE (Bordes et al., 2013), factorization-based ComplEx (Trouillon et al., 2016) and CNN-based ConvE (Dettmers et al., 2018). With the rise of deep learning, several DL-based methods have been proposed, such as ConvKB (Nguyen et al., 2018) and CompGCN (Vashisht et al., 2020). Balazevic et al. (Balazevic et al., 2019a) propose a linear model based on Tucker decomposition of the binary tensor representation of knowledge graph triples. RotatE (Sun et al., 2019), inspired by Euler’s identity, represents a relation as the rotation operation between the head and tail entities. Dihedral (Xu and Li, 2019) introduces rotation and reflection operations in dihedral symmetry group to construct the relation embeddings. Similar to the previous approaches, these models utilize high-dimensional embedding vectors while designing a new score function to better distinguish the triples.

### 6.2 Hyperbolic Embeddings

Hyperbolic geometry has recently drawn wide attention because of its potential to learn parsimonious representations of symbolic data by simultaneously capturing hierarchy and similarity (Nickel and Kiela, 2017; Sala et al., 2018; Le et al., 2019).

Recently, some researchers start to apply hyperbolic embedding in the KGE domain. Balažević et al. (Balazevic et al., 2019b) propose the MuRP model to embed KG triples in the Poincaré ball model of hyperbolic space using the Möbius matrix-vector multiplication and Möbius addition operations. Similarly, Kolyvakis et al. (Kolyvakis et al., 2020) extend the translational models by learning embeddings of KG entities and relations in the hyperbolic Poincaré-ball model. Sun et al.

(Sun et al., 2020) propose a hyperbolic relational graph neural network to capture knowledge associations for the KG entity alignment task. Chami et al. (Chami et al., 2020) employ rotation and reflection operations to replace the multiplication operation between the head entity and relation vectors, and propose a series of hyperbolic KGE models with trainable curvature, including RotH, RefH, and AttH.

Comparing with the existing hyperbolic KGE models, our model simplifies the hyperbolic calculations to improve computational efficiency while achieving competitive performance.

## 7 Conclusion

The recently proposed hyperbolic-based models achieve great prediction accuracy in low-dimensional knowledge graph embeddings, but require complicated calculations for hyperbolic embeddings. In this paper, we analyze the effective components in those models and propose a lightweight variant based on Euclidean calculations. After simplifying the Möbius operations in RotH, our proposed RotL model achieves a competitive performance, which saves half of the training time. Using a two-layer stacked transformation, we further propose Rot2L that outperforms the state-of-the-art RotH model in both prediction accuracy and training speed.

These positive results encourage us to explore further research activities in the future. We will theoretically analyze the effectiveness of flexible normalization in the low-dimensional KGE tasks. For the stacked transformations in Rot2L, we will explore multiple-layer architectures and evaluate more different transformation forms. Finally, we plan to apply our models on real-world knowledge graphs in different domains such as mobile healthcare, smart cities, and mobile e-Commerce.

## Acknowledgments

This research is supported by the National Natural Science Foundation in China (Grant: 61672128) and the Fundamental Research Fund for Central University (Grant: DUT20TD107). Quan Z. Sheng has been partially supported by Australian Research Council (ARC) Future Fellowship Grant FT140101247, and Discovery Project Grant DP200102298.

## References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019a. **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, Hong Kong, China. Association for Computational Linguistics.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019b. **Multi-relational poincaré graph embeddings**. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4465–4475.

Marcel Berger. 1987. *Geometry I*. Springer Berlin Heidelberg.

G. Birman and A. Ungar. 2001. The hyperbolic derivative in the poincaré ball model of hyperbolic geometry. *Journal of Mathematical Analysis and Applications*, 254:321–333.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94:233–259.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. **Translating embeddings for modeling multi-relational data**. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Samuel Broscheit, Kiril Gashteovski, Yanjie Wang, and Rainer Gemulla. 2020. **Can we predict new facts with open knowledge graph embeddings? a benchmark for open link prediction**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2296–2308, Online. Association for Computational Linguistics.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. **Low-dimensional hyperbolic knowledge graph embeddings**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. **Convolutional 2d knowledge graph embeddings**. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. **Hyperbolic neural networks**. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5350–5360.

Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. 2020. Hyperbolic knowledge graph embeddings for knowledge base completion. In *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, pages 199–214.

Matthew Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. **Infering concept hierarchies from text corpora via hyperbolic embeddings**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3231–3241, Florence, Italy. Association for Computational Linguistics.

George A. Miller. 1992. **WordNet: A lexical database for English**. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Maximilian Nickel and Douwe Kiela. 2017. **Poincaré embeddings for learning hierarchical representations**. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6338–6347.

Mrinmaya Sachan. 2020. **Knowledge graph embedding compression**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2681–2691, Online. Association for Computational Linguistics.

Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Ré. 2018. **Representation tradeoffs for hyperbolic embeddings**. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4457–4466. PMLR.

Zequin Sun, Muhan Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020. **Knowledge association with hyperbolic knowledge graph embeddings**. In *Proceedings of the 2020 Conference on Empirical*

*Methods in Natural Language Processing (EMNLP)*, pages 5704–5716, Online. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. **Rotate: Knowledge graph embedding by relational rotation in complex space**. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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, Beijing, China. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. **Complex embeddings for simple link prediction**. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

A. Ungar. 2001. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics With Applications*, 41:135–147.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. **Composition-based multi-relational graph convolutional networks**. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Kai Wang, Yu Liu, Qian Ma, and Quan Z. Sheng. 2021a. **Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings**. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1716–1726.

Kai Wang, Yu Liu, and Quan Z. Sheng. 2021b. Neighborhood intervention consistency: Measuring confidence for knowledge graph link prediction. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 2090–2096.

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*, 29:2724–2743.

Zhiwen Xie, Guangyou Zhou, Jin Liu, and Jimmy Xiangji Huang. 2020. **ReInceptionE: Relation-aware inception network with joint local-global structural information for knowledge graph embedding**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5929–5939, Online. Association for Computational Linguistics.

Canran Xu and Ruijiang Li. 2019. **Relation embedding with dihedral group in knowledge graph**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 263–272, Florence, Italy. Association for Computational Linguistics.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. **Quaternion knowledge graph embeddings**. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2731–2741.