Enhancing Hyperbolic Knowledge Graph Embeddings via Lorentz Transformations

Xiran Fan, Minghua Xu, Huiyuan Chen, Yuzhong Chen, Mahashweta Das, Hao Yang
Visa Research, Foster City, CA, USA
{xirafan, mixu, hchen, yuzchen, mahdas,haoyang}@visa.com

Abstract

Knowledge Graph Embedding (KGE) is a powerful technique for predicting missing links in Knowledge Graphs (KGs) by learning the entities and relations. Hyperbolic space has emerged as a promising embedding space for KGs due to its ability to represent hierarchical data. Nevertheless, most existing hyperbolic KGE methods rely on tangent approximation and are not fully hyperbolic, resulting in distortions and inaccuracies. To overcome this limitation, we propose LorentzKG, a fully hyperbolic KGE method that represents entities as points in the Lorentz model and represents relations as the intrinsic transformation—the Lorentz transformations between entities. We demonstrate that the Lorentz transformation, which can be decomposed into Lorentz rotation/reflection and Lorentz boost, captures various types of relations including hierarchical structures. Experimental results show that our LorentzKG achieves state-of-the-art performance[1].

1 Introduction

Knowledge graphs (KGs) (Dong et al., 2014), which comprise factual triples of the form *(head entity, relation, tail entity)*, are essential in question-answering (Huang et al., 2019), information extraction (Xiong et al., 2017), and recommendations (Wang et al., 2014; Chen et al., 2022a). Recent advancements related to KGs centered around knowledge graph embedding (KGE), which involves mapping entities and relations to some dedicated representation spaces, providing an effective tool for interpreting semantic meaning within KGs.

While many existing models utilize Euclidean space for modeling relation patterns in KGs (Bordes et al., 2013; Chen and Li, 2019; Yang et al., 2015; Sun et al., 2019), they face challenges in effectively modeling semantic hierarchies due to the limited expressive power of low-dimensional Euclidean space. Increasing the dimensions incurs high computational costs, especially given the large number of entities in KGs. Recent work has explored hyperbolic space, a non-Euclidean space with constant negative curvature, as a more effective and efficient embedding space for data with hierarchical structures (Nickel and Kiela, 2017; Chen and Li, 2020b; Sala et al., 2018; Wang et al., 2023; Zhao et al., 2024). Several hyperbolic KGE models have been proposed to represent KGs (Balazevic et al., 2019; Chami et al., 2020; Kolyvakis et al., 2020; Pan and Wang, 2021; Bai et al., 2021), demonstrating the advantages of hyperbolic space, particularly in the low-dimensional settings.

Most existing hyperbolic KGE models (Balazevic et al., 2019; Chami et al., 2020; Pan and Wang, 2021; Bai et al., 2021) rely on tangent space approximations to perform vector space operations, such as matrix-vector multiplication and vector addition. Tangent space is a vector space that approximates the manifold at a given base point. However, this approach is not fully hyperbolic since it involves lifting the data from hyperbolic space to tangent space for feature transformation and subsequently projecting it back to hyperbolic space. This projection introduces additional complexities and numerical instability, as it requires the calculation of a series of hyperbolic functions. Moreover, the tangent approximation of data that is located away from the base point is distorted and inaccurate, compromising the expressibility of hyperbolic space (Chen et al., 2022b; Fan et al., 2023).

To address above issues, it is essential to explore fully hyperbolic approaches that bypass the reliance on tangent space approximations and instead work directly within hyperbolic space. In this paper, we address this challenge by utilizing the *Lorentz model* (Ratcliffe, 2006) of hyperbolic space, and its isometry group, the positive *Lorentz group* (Gallier and Quaintance, 2012), to develop a fully hyperbolic KGE model, called LorentzKG.

[1]https://github.com/LorentzKG/LorentzKG

Findings of the Association for Computational Linguistics ACL 2024, pages 4575–4589

The property of the positive Lorentz group acting transitively on the Lorentz model enables the definition of matrix multiplication on the hyperbolic space as a linear operation—multiplying a group element of the Lorentz group, the *Lorentz transformation*, to a point on the Lorentz model. This linear approach avoids tangent approximation, and furthermore, benefits from the numerical stability exhibited by the Lorentz model (Nickel and Kiela, 2018; Fan et al., 2022), ensuring reliable and accurate calculations.

Our LorentzKG model follows a translation-based KGE methodology, where relation-specific transformations using Lorentz transformation are applied to the head and tail entities embedded in the Lorentz model. The plausibility of a given fact is measured by the hyperbolic distance between the transformed entities. The highlights of our LorentzKG are: **(i)** The transitive nature of Lorentz transformation on the Lorentz model ensures that the transformation is an *intrinsic linear operation* and guarantees that the transformed entity embedding remains a point in the Lorentz model; **(ii)** The nature of the Lorentz transformation, allowing for decomposition into various types of transformations, enables us to effectively capture diverse relational patterns observed in KGs, including symmetry/anti-symmetry, inversion, composition, and hierarchy (Sec 3.3); **(iii)** Different relational transformations on head and tail entities enable modeling of relation cardinalities (such as 1-to-1, 1-to-N, N-to-1, and N-to-N (Bordes et al., 2013; Sun et al., 2019)) (Sec 3.3); **(iv)** Experimental results demonstrate the superiority of our LorentzKG compared to several Euclidean and hyperbolic KGE models. In addition, we point out connections with prior translation-based Euclidean KGE models (Sec 3.4) and contrasts our LorentzKG with prevalent hyperbolic KGE models (Sec 3.2), underscoring our work's novelty.

Notations and Problem Setup. A KG is defined as a collection of factual triples $\mathcal{G} = \{(h, r, t)\}$, where head and tail entities $h, t \in \mathcal{E}$ and relation $r \in \mathcal{R}$ are taken from an entity set $\mathcal{E}$ and a relation set $\mathcal{R}$, respectively. The primary relation patterns in KGs include symmetry/antisymmetry, inversion, composition and hierarchy. The relationships can be categorized based on the cardinalities (of their heads or tails) as 1-to-1, 1-to-N, N-to-1, and N-to-N (Bordes et al., 2013). In KGE models, the objective is to map entities $h, t \in \mathcal{E}$ to low-dimensional embeddings $\boldsymbol{h}, \boldsymbol{t} \in \mathcal{U}^{d_\mathcal{E}}$ and relations $r \in \mathcal{R}$ to their embeddings $\boldsymbol{r} \in \mathcal{U}^{d_\mathcal{R}}$, where $\mathcal{U}$ is a chosen representation space, e.g, Euclidean space $\mathbb{R}$, and $d_\mathcal{E}$ and $d_\mathcal{R}$ are the dimensionalities of entity and relation embeddings, respectively. The learning of embeddings in KGE involves optimizing a score function $f_{\boldsymbol{r}}(\boldsymbol{h}, \boldsymbol{t}) : \mathcal{U}^{d_\mathcal{E}} \times \mathcal{U}^{d_\mathcal{R}} \times \mathcal{U}^{d_\mathcal{E}} \to \mathbb{R}$, which evaluates the validity of each triple. The embeddings of entities and relations are trained by optimizing the score function, with the objective of maximizing the scores of valid triples.

2 Preliminaries

Hyperbolic space is a Riemannian manifold with constant negative curvature. There are five isometric models of hyperbolic space (Cannon et al., 1997; Ratcliffe, 2006), including the Lorentz model $\mathbb{L}$ and the Poincaré ball model $\mathbb{B}$ (see Appendix B). We select the *Lorentz model* as the representation space in LorentzKG due to its nature as a homogeneous space and its intrinsic characteristic of the positive *Lorentz group* being its isometry group. In this section, we provide a brief overview of these concepts and important geometric notions in hyperbolic space that are relevant to our work.

2.1 Lorentz Model and Lorentz Transformation

Lorentzian Space. The $(n+1)$-dimensional *Lorentzian space* $\mathbb{R}^{1,n}$ is the Euclidean space $\mathbb{R}^{n+1}$ equipped with a *non-positive-definite* bilinear form:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_\mathbb{L} = -x_0 y_0 + x_1 y_1 + \cdots + x_n y_n, \quad (1)$$

where $\boldsymbol{x} = [x_0, x_1, \ldots, x_n]^T, \boldsymbol{y} = [y_0, y_1, \ldots, y_n]^T \in \mathbb{R}^{n+1}$. The bilinear form $\langle \cdot, \cdot \rangle_\mathbb{L}$ is called *Lorentzian inner product*.

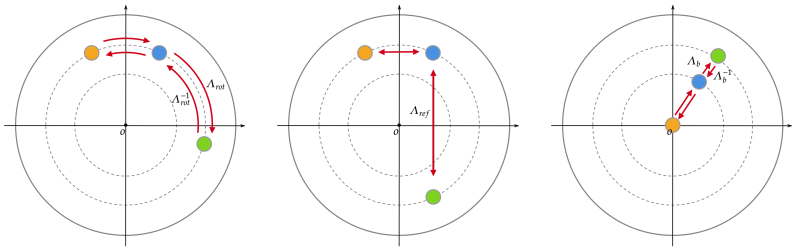
Lorentz Model. The n-dimensional *Lorentz model* $\mathbb{L}^n$ is a submanifold in $\mathbb{R}^{1,n}$ defined as:

$$\mathbb{L}^n := \{\boldsymbol{x} \in \mathbb{R}^{n+1} : \langle \boldsymbol{x}, \boldsymbol{x} \rangle_\mathbb{L} = -1, x_0 > 0\}. \quad (2)$$

The Lorentz model is the upper sheet of the two-sheeted n-dimensional hyperboloid in $\mathbb{R}^{1,n}$.

Squared Lorentzian Distance. The squared Lorentzian distance is a modified distance metric used in the hyperbolic space (Ratcliffe, 2006). For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{L}^n$, the squared Lorentzian distance is:

$$d_\mathbb{L}^2 = \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle_\mathbb{L} = -2 - 2\langle \boldsymbol{x}, \boldsymbol{y} \rangle_\mathbb{L}. \quad (3)$$



(a) Lorentz rotation (b) Lorentz reflection (c) Lorentz boost

Figure 1: Illustration (in the Poincaré ball model $\mathbb{B}^2$) of: (a) Lorentz rotation with rotation matrix $\mathbf{\Lambda}_{rot}$ and the inverse rotation matrix $\mathbf{\Lambda}_{rot}^{-1}$; (b) Lorentz reflection with reflection matrix $\mathbf{\Lambda}_{ref}$ (the inverse reflection matrix is $\mathbf{\Lambda}_{ref}$ itself); (c) Lorentz boost with boost matrix $\mathbf{\Lambda}_b$ and the inverse boost matrix $\mathbf{\Lambda}_b^{-1}$.

Lorentz Transformation. Linear isometries that preserve the Lorentzian inner product between every pair of points in the Lorentzian space are called Lorentz transformations. Specifically, a map $\phi : \mathbb{R}^{1,n} \to \mathbb{R}^{1,n}$ is a Lorentz transformation if $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle_{\mathbb{L}} = \langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{L}}, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{L}^n$.

Lorentz Group. All Lorentz transformations form a group under composition. This group is called the *Lorentz group*, denoted by $\mathbf{O}(1, n)$. Let $J_n = \text{diag}(-1, \mathbf{1}_n)$ where $\mathbf{1}_n$ is n-dimensional vector with all entries being 1 and diag$(\cdot)$ denotes a diagonal matrix, the Lorentz group is defined as:

$$\mathbf{O}(1,n) \coloneqq \{A \in GL(n+1,\mathbb{R}) : AJ_nA^T = A^TJ_nA = J_n\}, \tag{4}$$

where $GL(n+1, \mathbb{R})$ is the general linear group of $(n+1) \times (n+1)$-invertible matrices over $\mathbb{R}$.

Positive Lorentz Group. The positive Lorentz group is a subgroup of $\mathbf{O}(1, n)$ defined as $\mathbf{O}^+(1,n) \coloneqq \{A \in \mathbf{O}(1,n) : a_{11} > 0\}$. The group acts transitively on $\mathbb{L}^n$ through the group action $\boldsymbol{x} \mapsto A\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{L}^n$ and $A \in \mathbf{O}^+(1,n)$. Specifically, $A\boldsymbol{x} \in \mathbb{L}^n$, $\forall \boldsymbol{x} \in \mathbb{L}^n$ and $\forall A \in \mathbf{O}^+(1,n)$. As a result, $\mathbf{O}^+(1,n)$ is the isometry group of the Lorentz model $\mathbb{L}^n$.

Lorentz Rotation/Reflection and Lorentz Boost. A Lorentz transformation $A \in \mathbf{O}^+(1,n)$ can be decomposed using a polar decomposition (Moretti, 2002) and expressed as:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} c & \boldsymbol{v}^T \\ \boldsymbol{v} & \sqrt{I_n + \boldsymbol{v}\boldsymbol{v}^T} \end{bmatrix}, \tag{5}$$

where I_n is the identity matrix, $R \in \mathbf{O}(n)$, the orthogonal group (i.e., $R^TR = I_n$), $\boldsymbol{v} \in \mathbb{R}^n$, and $c = \sqrt{\|\boldsymbol{v}\|^2 + 1}$. The first component is called a Lorentz rotation Fig. 1(a) if $det(R) = 1$, or a Lorentz reflection Fig. 1(b) if $det(R) = -1$. Lorentz rotation/reflection is space rotation/reflection as it does not change the time axis. The second component is a Lorentz boost Fig. 1(c), where $\boldsymbol{v}$ is the velocity vector that describes the magnitude and direction of the boost.

3 Methodology

Here we present our proposed LorentzKG which uses the Lorentz model for entity embeddings and Lorentz transformation for relation embeddings in KGs. The score function measures triple plausibility using the squared Lorentzian distance between transformed entity embeddings, along with a bias term. We also discuss LorentzKG's capability to model KG relations, highlighting its potential in capturing diverse relational patterns.

3.1 LorentzKG

In our approach, entities are represented as points in the *Lorentz model*, i.e., $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{L}^k$, while relations are represented as geometric transformations applied to entity embeddings. Specifically, relations are represented as *Lorentz transformations* embedded in the *positive Lorentz group* $\mathbf{O}^+(1,k)$. Our approach brings forth the following significant benefits: **(i)** We can ensure that the transformed entities remain within hyperbolic space since Lorentz transformations are linear isometries of the Lorentz model. **(ii)** The linear operation is simple and stable to compute. **(iii)** The use of the Lorentz transformations allows for multiple transformations, including rotation, reflection, inversion, and translation as it is the isometry group of the Lorentz model, which will be discussed later. Next, we present our simple yet effective score function of LorentzKG.

Score Function. The score function in hyperbolic KGE models is typically defined as the squared hyperbolic distance between transformed entities in the hyperbolic space, followed by scalar biases for entities and a margin (Balažević et al., 2019; Chami et al., 2019; Chen et al., 2022b). Building upon this framework, we design the score function in our LorentzKG as follows:

$$f_r(\boldsymbol{h}, \boldsymbol{t}) = -d^2_{\mathbb{L}}(\boldsymbol{\Lambda}_{\boldsymbol{r},1}\boldsymbol{h}, \boldsymbol{\Lambda}_{\boldsymbol{r},2}\boldsymbol{t}) + b_{\boldsymbol{h}} + b_{\boldsymbol{t}} + \delta, \quad (6)$$

where $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{L}^k$ are entity embeddings in Lorentz model, $\boldsymbol{\Lambda}_{\boldsymbol{r},1}, \boldsymbol{\Lambda}_{\boldsymbol{r},2} \in \mathbf{O}^+(1,k)$ are relation matrices, $b_{\boldsymbol{h}}, b_{\boldsymbol{t}} \in \mathbb{R}$ are scalar biases of head entity $\boldsymbol{h}$ and tail entity $\boldsymbol{t}$, respectively, $\delta \in \mathbb{R}$ is a margin hyperparameter and $d^2_{\mathbb{L}}$ is the squared Lorentz distance function given in Eq. 3. The terms $\boldsymbol{\Lambda}_{\boldsymbol{r},1}\boldsymbol{h}$ and $\boldsymbol{\Lambda}_{\boldsymbol{r},2}\boldsymbol{t}$ are referred to as the transformed (by relation) head and tail entities, respectively.

The geometric interpretation of the score function given in Eq. 6 is straightforward. The head entity h and the tail entity t are embedded in the Lorentz model, and a Lorentz transformation, which is the intrinsic operation on the Lorentz model, is used to model the relation r. This transformation determines the projected position of the entity embeddings adjusted by the relation. In a valid fact (h, r, t), the transformed h and t (by relation) should be brought closer to each other, as measured by the hyperbolic distance. Additionally, the entity-specific biases $b_{\boldsymbol{h}}$ and $b_{\boldsymbol{t}}$ can be interpreted as the sphere of influence of each entity, as suggested by (Balazevic et al., 2019). In practice, we enforce the constraints on the bias term $b_{\boldsymbol{h}}, b_{\boldsymbol{t}} \in (-1, 1)$ following (Lin et al., 2015).

3.2 Comparison with Existing Hyperbolic KGE Models

In the context of entity transformation, a comparison between LorentzKG and other hyperbolic KGE models reveals distinct approaches. For a given head entity $\boldsymbol{h}$, we call the relation-specified head entity transformed-$\boldsymbol{h}$ and analyze its distinctions across various hyperbolic KGE models.

In LorentzKG, the process involves applying a Lorentz transformation directly to entities represented in the Lorentz model ($\boldsymbol{h} \in \mathbb{L}^k$):

$$\text{transformed-}\boldsymbol{h} = \boldsymbol{\Lambda}_r \boldsymbol{h} \quad (7)$$

where $\boldsymbol{\Lambda}_{\boldsymbol{r}} \in \mathbf{O}^+(1,k)$ a Lorentz transformation. Consequently, transformed-$\boldsymbol{h}$ naturally exists as a point in the hyperbolic space $\mathbb{L}^k$.

In contrast, HyboNet (Chen et al., 2022b) proposed a *general* hyperbolic linear layer for entiry transformation ($\boldsymbol{h} \in \mathbb{L}^k$):

$$\text{transformed-}\boldsymbol{h} = \begin{bmatrix} \sqrt{\|\phi(\boldsymbol{W_r h})\| + 1} \\ \phi(\boldsymbol{W_r h}) \end{bmatrix}_{k+1 \times k+1} \quad (8)$$

where $\boldsymbol{W_r} \in \mathbb{R}^{k \times k}$ is an arbitrary matrix and $\phi(\cdot)$ is an operation function, which includes a composition of operations such as dropout, normalization, and non-linear activation. To ensure the maintenance of transformed-$\boldsymbol{h}$ within the hyperbolic space, an additional scalar element is computed according to Eq. 2 and appended to the output $\phi(\boldsymbol{W_r h})$. While Chen et al. (2022b) purports to present a fully hyperbolic method, it is noteworthy that the general hyperbolic linear layer employed in their implementation *does not* conform to the hyperbolic nature (Chen et al., 2022b, §3.1). Their implementation involves moving embeddings out of the manifold and then projecting them back, making it non-intrinsic. In contrast, we introduce simple and scalable entity transformations that consistently keep the learning trajectory on the manifold.

Moreover, AttH (Chami et al., 2020) employs a block-wise rotation/reflection matrix to transform entities in the Poincaré ball model ($\boldsymbol{h} \in \mathbb{B}^k$):

$$\text{transformed-}\boldsymbol{h} = \begin{bmatrix} \boldsymbol{G}_{\boldsymbol{r}}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{G}_{\boldsymbol{r}}^{k/2} \end{bmatrix}_{k \times k} \boldsymbol{h} \quad (9)$$

where $\boldsymbol{G}_{\boldsymbol{r}}^i = \begin{bmatrix} \cos\theta_i & \mp\sin\theta_i \\ \sin\theta_i & \pm\cos\theta_i \end{bmatrix}$ represents a 2×2 rotation/reflection matrix. Our LorentzKG distinguishes itself by its capacity to execute a $k \times k$ rotation/reflection within the k-dimensional hyperbolic space. This capability contrasts AttH's limitation to a diagonal 2×2 rotation/reflection, thereby constraining its expressive power.

3.3 Capability in Modeling Relation Patterns

The Lorentz transformations offer a powerful approach for modeling various relational patterns. Our model captures symmetric/anti-symmetric relations through Lorentz rotations/reflections, as these transformations correspond to Euclidean rotations/reflections. Also, our model captures composition and inversion since all Lorentz transformations form the Lorentz group. Please refer to Appendix D for proofs.

Moreover, our method enables the modeling of hierarchical relations using Lorentz boosts. The idea in modeling hierarchical relations in KGs is to embed the entities at the same hierarchical level with similar radii (in the Poincaré ball model). In contrast to existing hierarchy-aware KGE models such as HAKE (Zhang et al., 2020) (Euclidean) and HBE (Pan and Wang, 2021) (hyperbolic), where the score function is a weighted average of modulus distance and phase distance in polar coordinates, our method provides a more streamlined solution that avoids the need to introduce separate distance measures for radii and angles. By applying a Lorentz boost, the transformed entity embedding experiences a change in radii (in the Poincaré ball model), thereby effectively capturing hierarchy.

Furthermore, LorentzKG effectively models relation cardinalities by applying different Lorentz transformations to both the head and tail entities. This implementation allows LorentzKG to capture and represent relation cardinalities, facilitating a comprehensive understanding of the underlying relationships between entities.

3.4 Connection to Euclidean Translation-based KGE Models

Our LorentzKG can be regarded as a natural generalization of Euclidean translation-based models to hyperbolic space. For example, the score function in STransE (Nguyen et al., 2016) is $f_{\boldsymbol{r}}(\boldsymbol{h},\boldsymbol{t}) = \|\boldsymbol{M}_{r,1}\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{M}_{r,2}\boldsymbol{t}\|_2$ where $\boldsymbol{h},\boldsymbol{t},\boldsymbol{r} \in \mathbb{R}^k$ and $\boldsymbol{M}_{r,i} \in \mathbb{R}^{k\times k}$. If we reformulate the score function by writing entity embeddings in homogeneous coordinates, the score is:

$$
\begin{aligned}
&f_{\boldsymbol{r}}(\boldsymbol{h},\boldsymbol{t}) = \|\hat{\boldsymbol{M}}_{r,1}\hat{\boldsymbol{h}} - \hat{\boldsymbol{M}}_{r,2}\hat{\boldsymbol{t}}\|_2, \\
&\hat{\boldsymbol{M}}_{r,1} = \begin{bmatrix} \boldsymbol{M}_{r,1} & 0 \\ 0 & \boldsymbol{r} \end{bmatrix}, \quad \hat{\boldsymbol{h}} = \begin{bmatrix} \boldsymbol{h} \\ 1 \end{bmatrix}, \\
&\hat{\boldsymbol{M}}_{r,2} = \begin{bmatrix} \boldsymbol{M}_{r,1} & 0 \\ 0 & 0 \end{bmatrix}, \quad \hat{\boldsymbol{t}} = \begin{bmatrix} \boldsymbol{t} \\ 1 \end{bmatrix}.
\end{aligned} \tag{10}
$$

The score function in Eq. 6 of LorentzKG has the same form as that in STransE, making our method a generalization of STransE to hyperbolic space. Note that in STransE, the matrix embeddings of relations are arbitrary, implying that the representation space for relations forms the *affine group*. This arbitrariness frequently results in suboptimal performance due to over-parametrization, a limitation also observed in RESCAL (Nickel et al., 2011). In contrast, our approach in LorentzKG involves employing the *Lorentz group* to curtail the degree of freedom in embedding relations while ensuring that the transformed entities remain points within the hyperbolic space.

3.5 Training and Optimization

Loss function. We follow the setup in (Balazevic et al., 2019; Chen et al., 2022b) to train our proposed LorentzKG model. We construct K negative samples for each triple (h,r,t) by corrupting either the tail entity (h,r,t') or the head entity (h',r,t) with a randomly chosen new entity from the entity set $\mathcal{E}$. Let $y \in \{0,1\}$ denote the true label indicating the validity of a triplet. We apply the sigmoid function $\sigma(\cdot)$ to the scores $f_{\boldsymbol{r}}(\boldsymbol{h},\boldsymbol{t})$ to calculate probabilities for triples and minimize the binary cross entropy loss:

$$
L(y,p) = -\frac{1}{N}\sum_{i=1}^{N}[\log(p_i) + \sum_{k=1}^{K}\log(1-p'_{(i,k)})] \tag{11}
$$

where p_i and $p'_{(i,k)}$ are the probabilities for correct and corrupted triples, respectively. N is the number of training samples and K is the number of negative samples for a positive sample.

Optimization. Instead of directly updating the relation embeddings $\boldsymbol{\Lambda}_{\boldsymbol{r},i}$, a Lorentz transformation is decomposed using Eq. 5 into a composition of a Lorentz rotation/reflection and a Lorentz boost, which are then sequentially updated during optimization. Specifically, the Lorentz rotation/reflection is learned by learning a rotation (orthogonal) matrix R, while the Lorentz boost is learned by learning a real vector $\boldsymbol{v}$. We implemented R as the Householder matrix (Uhlig, 2001) to enforce its orthogonality. We use *Riemannian Adam* (rAdam) (Becigneul and Ganea, 2019) to optimize the proposed model.

4 Experiments

Dataset. We evaluate our proposed model on two widely used knowledge graph datasets - WN18RR (Dettmers et al., 2018) and FB15k-237 (Toutanova et al., 2015). Some statistics of these two knowledge graph datasets are summarized in Table. 1. $\xi_{\mathcal{G}}$ is the global graph curvature that measures the level of hierarchy in graphs. The lower $\xi_{\mathcal{G}}$ indicates the more tree-like the KG is (Chami et al., 2020). More experiments are included in Sec G.

Evaluations. We evaluate the performance of link prediction in the filtered setting by ranking

Dataset	#entities	#relations	#training	#valid	#test	$\xi_{\mathcal{G}}$
WN18RR (Dettmers et al., 2018)	40,943	11	86,835	3,034	3,134	-2.54
FB15k-237 (Toutanova et al., 2015)	14,451	237	272,115	17,535	20,466	-0.65

Table 1: Statistics of knowledge graph datasets. $\xi_{\mathcal{G}}$ measures the hierarchy in graphs (Chami et al., 2020).

	WN18RR					FB15k-237				
Model	#Dim	MRR	H@10	H@3	H@1	#Dim	MRR	H@10	H@3	H@1
TansE	180	22.7	50.6	38.6	3.5	200	28.0	48.0	32.1	17.7
DistMult	270	41.5	48.5	43.0	38.1	200	19.3	35.3	20.8	11.5
ComplEx	230	43.2	50.0	45.2	39.6	200	25.7	44.3	29.3	16.5
ConvE	120	43.5	50.0	44.6	40.1	200	30.4	49.0	33.5	21.3
RotatE	1000	47.3	55.3	48.8	43.2	1024	30.1	48.5	33.1	21.0
QuatE	100	48.1	56.4	50.0	43.6	100	31.1	49.5	34.2	22.1
TuckER	200	47.0	52.6	48.2	44.3	200	35.8	54.4	39.4	25.6
HAKE	1000	<u>49.7</u>	<u>58.2</u>	<u>51.6</u>	45.2	1000	34.6	54.2	38.1	25.0
GIE	300	49.1	57.5	50.5	45.2	800	36.2	55.2	<u>40.1</u>	27.1
M^2GNN	200	48.5	57.2	49.8	44.4	200	<u>36.2</u>	<u>56.5</u>	39.8	<u>27.5</u>
HyperKG	100	41.1	50.0	-	-	100	28.0	45.2	-	-
UltraE	32	48.8	55.8	50.3	44.0	32	33.8	51.4	36.3	24.7
MuRP	32	46.5	54.4	48.4	42.0	32	32.3	50.1	25.3	23.5
RotH	32	47.3	55.3	49.0	42.8	32	31.4	49.7	34.6	22.3
AttH	32	46.6	55.1	48.4	41.9	32	32.4	50.1	35.4	23.6
ConE	32	47.1	53.7	48.6	43.6	32	27.6	46.0	30.6	18.6
HyboNet	32	48.9	55.3	50.2	<u>45.5</u>	32	33.4	51.6	36.5	24.4
LorentzKG	32	**50.2**	**58.9**	**52.3**	**45.6**	32	**38.4**	**57.9**	**42.2**	**28.7**

Table 2: Embedding dimensions and performances of various KGE models on WN18RR and FB15k-237 datasets, with most results obtained from (Chen et al., 2022b) and original papers (QuatE (Zhang et al., 2019), TuckER (Balažević et al., 2019), HAKE (Zhang et al., 2020), GIE (Cao et al., 2022), HyperKG (Kolyvakis et al., 2020), UltraE (Xiong et al., 2022) and ConE (Bai et al., 2021)). Results of ConE on FB15k-237 are obtained from our own implementation. The best results are boldfaced, and the second best ones are underlined.

test triples against all other candidate triples that do not appear in the training, validation, or test sets. To generate these candidates, we corrupt either the subject or object of the original triple to form (h', r, t) or (h, r, t'). Evaluation is conducted using standard measures for these datasets, including Mean Reciprocal Rank (MRR) and Hits at N (H@N), where $N = 1, 3$ or 10.

Baselines. We compare our LorentzKG with several baseline methods. For Euclidean KGE models, we include TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016) ConvE (Dettmers et al., 2018), RotatE (Sun et al., 2019), QuatE (Zhang et al., 2019), TuckER (Balažević et al., 2019) and HAKE (Zhang et al., 2020). Regarding hyperbolic KGE models , we consider GIE (Cao et al., 2022), M^2GNN (Wang et al., 2021), UltraE (Xiong et al., 2022), HyperKG (Kolyvakis et al., 2020), MuRP (Balazevic et al., 2019), RotH and AttH (Chami et al., 2020), ConE (Bai et al., 2021) and HyboNet (Chen et al., 2022b). For the hyperbolic KGE model, we mainly present the results in the lower dimensions (dim = 32).

Implementation Details. The implementation of our LorentzKG is based on PyTorch, with the corresponding PyTorch-based pseudocode presented in Section E. All experiments are conducted on a NVIDIA Tesla Nvidia GPU A100 40GB machine. We employ random initialization on a manifold using the Python package Geoopt (Kochurov et al., 2020). To determine the optimal hyperparameters for our method, we conduct a grid search for learning rate, negative sample size, batch size, and margin. The best hyperparameters for each dataset are reported as follows: {WN18RR: 0.05, 200, 512, 1.08}, {FB15k-237:0.05,100,512,1.15}.

Overall Performance. We evaluate our approach in the low-dimensional setting for d = 32. The experimental results presented in Table 2 clearly illustrate the superior performance of our proposed LorentzKG model compared to existing methods in both Euclidean and hyperbolic spaces. On WN18RR, HAKE shows comparable results but with a 1000-dimensional representation space. The

Model	WN18RR				FB15k-237			
	MRR	H@10	H@3	H@1	MRR	H@10	H@3	H@1
LorentzKG-h	49.8	58.5	52.1	45.3	36.6	54.0	39.6	28.0
LorentzKG-t	50.0	58.6	52.2	45.4	36.7	54.1	39.7	28.0
LorentzKG-r	49.7	58.0	51.9	40.2	37.5	57.1	41.6	28.6
LorentzKG-b	36.1	55.9	42.8	25.3	24.3	44.8	26.8	14.5
LorentzKG	50.2	58.9	52.3	45.6	38.4	57.9	42.2	28.7

Table 3: Ablations on (i) head (-h) and tail (-t) transformation; and (ii) rotation/reflection (-r) and boost (-b) for our LorentzKG.

	WN18RR	FB15k-237
RotatE	40.95M	20.32M
MuRP	32.76M	5.82M
GIE	17.65M	3.41M
LorentzKG	**1.39M**	**1.42M**

Table 4: Comparison of the number of parameters.

reason may lie in the fact that hierarchy is one of the main relation patterns in WN18RR, as also evidenced by the lower $\xi_{\mathcal{G}}$. Furthermore, considering the low-dimensional embedding space, the overall performance of hyperbolic KGE models is comparable, aligning with the understanding that hyperbolic space is efficient in representing hierarchical structures. Thus, our standout performance establishes the superiority of our approach in modeling various relation patterns, including hierarchy. FB15k-237 exhibits a larger number of relations but fewer entities compared to WN18RR, resulting in a more complex graph structure. The utilization of different geometric structures in GIE and M^2GNN enhances their capability to effectively handle such intricate structures, leading to improved performance. However, LorentzKG outperforms M^2GNN (the second-best model across most metrics) with a (relative improvement) 6.1% higher MRR, a 2.5% higher H@10, a 6.0% higher H@3, and a 4.4% higher H@1, respectively. This improvement is mainly attributed to the expressivity offered by the various types of Lorentz transformations utilized in our approach. As a result, our approach demonstrates wide applicability in handling complex knowledge graph structures.

Model Capacity. We compare the number of parameters of several KGE models: RotatE, MuRP, and GIE (values are sourced from (Cao et al., 2022)), and our proposed LorentzKG. As illustrated in Table. 4, it is evident that our LorentzKG achieves superior performance with a significantly reduced parameter count, attributed to our exploration of low-dimensional hyperbolic space. Please refer to Sec H for more analysis on model capacity.

Ablation Studies. We conducted two types of ablation studies on our model. First, to assess the effects of applying the transformation exclusively to either the head entity (LorentzKG-h) or to the tail entity (LorentzKG-t). As demonstrated in Table 3, both variants of LorentzKG exhibited comparable results. However, through the synergistic utilization of both transformations, LorentzKG achieved superior performance by affording enhanced flexibility in capturing relational patterns. Second, to examine the impact of different types of Lorentz transformations (Lorentz rotation/reflection and Lorentz boost) in relational modeling, we compared our LorentzKG with its two variants: LorentzKG-r, which applies only Lorentz rotation/reflection to transform entities, and LorentzKG-b, which applies only Lorentz boost. The experimental results, presented in Table. 3, indicate that LorentzKG-r performs comparably to LorentzKG, as the majority of relation information is encoded by Lorentz rotation/reflection, which possesses a higher number of parameters compared to Lorentz boost. However, by leveraging the benefits of both types of transformations, LorentzKG achieves a more comprehensive representation of relational patterns and captures a wider range of semantic relationships, as evidenced by its superior performance in the table.

Embedding Dimension. To investigate the influence of embedding dimension, we perform experiments on WN18RR and FB15k-237 datasets using various settings $\{32, 64, 100, 200\}$, as shown in Table. 5. Our observations reveal improved performance (in most metrics) with increasing embedding dimensions, demonstrating that LorentzKG excels in both low and high dimensions.

Model	WN18RR					FB15k-237				
	#Dim	MRR	H@10	H@3	H@1	#Dim	MRR	H@10	H@3	H@1
LorentzKG	32	50.2	58.9	52.3	45.6	32	38.4	57.9	42.2	28.7
LorentzKG	64	51.4	59.2	52.7	44.6	64	38.7	58.1	42.6	29.1
LorentzKG	100	52.1	60.2	53.0	45.2	100	38.9	58.6	43.3	29.5
LorentzKG	200	52.4	61.3	54.2	46.0	200	39.2	59.0	43.8	29.3

Table 5: Embedding dimensions and performances of LorentzKG on WN18RR and FB15k-237 datasets.

5 Related Work

We review some KGE models, highlighting their differences in two aspects: the selection of the embedding space and the score function (Table. A2).

5.1 Euclidean Embeddings

Bilinear models, such as RESCAL (Nickel et al., 2011), represent relations as linear transformations acting on entity vectors. DistMult (Yang et al., 2015) is a special case of RESCAL that reduces the number of parameters per relation, while ComplEx (Trouillon et al., 2016) extends DistMult into the complex domain to model antisymmetry.

Translation-based models including TransE (Bordes et al., 2013) and its variants, treat relations as translations from head to tail entity embeddings, e.g., $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$. However, they cannot capture specific logical properties such as symmetry and cannot model relations with higher cardinalities if the translation is only applied to the head entity, e.g., TransE. Rotation-based models define relations as rotations in complex space, RotatE (Sun et al., 2019), or hypercomplex space, QuatE (Zhang et al., 2019) to enable expressive semantic matching between head and tail entities. While these models can capture important relation patterns in KGs, they are limited in modeling hierarchy and usually suffer from high complexity (Bordes et al., 2013; Chen and Li, 2020a; Sun et al., 2019; Lai et al., 2023; Jin et al., 2024; Moosaei et al., 2022).

5.2 Hyperbolic Embeddings

As HAKE (Zhang et al., 2020) demonstrates, relations in KGs often exhibit semantic hierarchies. For instance, within the triplets *(Alice, _is_friend_of, Bob)* and *(Bob, _is_supervisor_of, Chris)* (Cai et al., 2018), entities *Alice* and *Bob* share the same hierarchical level, whereas *Chris* is considered at a different hierarchical level. HAKE employes the polar coordinate system in Euclidean space to model these hierarchies, by assuming that entities at different hierarchical levels have distinct magnitudes in polar coordinates. Recently, hyperbolic space has been increasingly used for KGs due to its effectiveness in capturing hierarchical structures.

In hyperbolic KGE models, the likelihood of a fact is assessed by measuring the hyperbolic distance between the translated embeddings of its entities in the hyperbolic space. MuRP (Balazevic et al., 2019) and HyperKG (Kolyvakis et al., 2020) are two early attempts in the development of translation-based hyperbolic KGE methods on the *Poincaré ball model*. MuRP employs Möbius matrix-vector multiplication and Möbius addition (Ungar, 2005), which are generalizations of Euclidean operations in tangent space, to translate entity embeddings. HyperKG utilizes Euclidean addition and circular permutation. However, these operations do not completely reflect hyperbolic geometry. AttH (Chami et al., 2019) uses a combination of block-wise hyperbolic reflections/rotations, Eq. 9, and attention modules to model symmetric/antisymmetric relational patterns. UltraE (Xiong et al., 2022) adopts the same approach but uses *ultrahyperbolic manifold* as embedding space. HBE (Pan and Wang, 2021) adopts the idea of using a polar coordinate system to capture hierarchical structures to the *extended Poincaré ball model*. ConE (Bai et al., 2021) uses hyperbolic cones to represent entities and models relations as transformations between the cones. However, Möbius operations are applied in both HBE and ConE. HyboNet (Chen et al., 2022b) operates on the *Lorentz model*, where the head entity is transformed using a Lorentz linear transformation layer as described in Eq. 8. However, none of the existing hyperbolic KGE models apply transformations to both the head and tail entities, i.e., the hyperbolic distance is measured for the transformed head entity and the tail entity, which prevents these models from capture relation cardinalities, similar to the case of STransE (Nguyen et al., 2016).

Several recent studies have investigated combining other topological structures (e.g., hypersphere) with hyperbolic space in KGE. Notable examples

include M^2GNN (Wang et al., 2021) and GIE (Cao et al., 2022) where tangent space operations are applied in these models.

6 Conclusions

Here we introduce LorentzKG, a fully translation-based hyperbolic KGE model that utilizes Lorentz transformations. LorentzKG enables linear translation of entity embeddings and facilitates the modeling of various relation patterns in KGs. By applying different transformations to the head and tail entities, LorentzKG is able to capture relation cardinalities. Experimental results show the superiority of LorentzKG over existing models, highlighting its enhanced expressivity and modeling capabilities, particularly in low-dimensional settings.

7 Limitations

This study primarily concentrates on leveraging hyperbolic geometry to improve hyperbolic knowledge graph embeddings, with a particular emphasis on low-dimensional embeddings. Our comparative analysis was confined to translation-based KGE models and hyperbolic models. This particular focus might potentially narrow the breadth of our findings. Future research endeavors should consider diversifying the scope to include more diverse settings and larger-scale data to further validate and extend the applicability of our proposed methodologies.

References

Yushi Bai, Zhitao Ying, Hongyu Ren, and Jure Leskovec. 2021. Modeling heterogeneous hierarchies with relation-specific hyperbolic cones. In *Advances in Neural Information Processing Systems*.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. Multi-relational poincaré graph embeddings. *Advances in Neural Information Processing Systems*.

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*.

Gary Becigneul and Octavian-Eugen Ganea. 2019. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*.

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*.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Twenty-fifth AAAI conference on artificial intelligence*.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*.

James W. Cannon, William J. Floyd, Richard Kenyon, and Walter R. Parry. 1997. *Hyperbolic Geometry*. MSRI Publications.

Zongsheng Cao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. 2022. Geometry interaction knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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*.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Advances in neural information processing systems*.

Huiyuan Chen and Jing Li. 2019. Adversarial tensor factorization for context-aware recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*.

Huiyuan Chen and Jing Li. 2020a. Learning data-driven drug-target-disease interaction via neural tensor network. In *International Joint Conference on Artificial Intelligence*.

Huiyuan Chen and Jing Li. 2020b. Neural tensor model for learning multi-aspect factors in recommender systems. In *International Joint Conference on Artificial Intelligence*.

Huiyuan Chen, Xiaoting Li, Kaixiong Zhou, Xia Hu, Chin-Chia Michael Yeh, Yan Zheng, and Hao Yang. 2022a. Tinykg: Memory-efficient training framework for knowledge graph neural recommender systems. In *Proceedings of the 16th ACM Conference on Recommender Systems*.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022b. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Xiran Fan, Chun-Hao Yang, and Baba C Vemuri. 2022. Nested hyperbolic spaces for dimensionality reduction and hyperbolic nn design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xiran Fan, Chun-Hao Yang, and Baba C. Vemuri. 2023. Horospherical decision boundaries for large margin classification in hyperbolic space. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jean Gallier and Jocelyn Quaintance. 2012. Notes on differential geometry and lie groups. *University of Pennsylvannia*.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the twelfth ACM international conference on web search and data mining*.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. In *International conference on machine learning*.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. 2020. Geoopt: Riemannian optimization in pytorch.

Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. 2020. Hyperbolic knowledge graph embeddings for knowledge base completion. In *European Semantic Web Conference*. Springer.

Vivian Lai, Huiyuan Chen, Chin-Chia Michael Yeh, Minghua Xu, Yiwei Cai, and Hao Yang. 2023. Enhancing transformers without self-supervised learning: A loss landscape perspective in sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*.

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*.

Maryam Moosaei, Yusan Lin, Ablaikhan Akhazhanov, Huiyuan Chen, Fei Wang, and Hao Yang. 2022. Outfitgan: Learning compatible items for generative fashion outfits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Valter Moretti. 2002. The interplay of the polar decomposition theorem and the lorentz group. *ArXiv*.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*.

Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*.

Zhe Pan and Peng Wang. 2021. Hyperbolic hierarchy-aware knowledge graph embedding for link prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

John G Ratcliffe. 2006. *Foundations of Hyperbolic Manifolds*. Springer.

Tara Safavi and Danai Koutra. 2020. Codex: A comprehensive knowledge graph completion benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. 2018. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*.

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

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*.

Frank Uhlig. 2001. Constructive ways for generating (generalized) real orthogonal matrices as products of (generalized) symmetries. *Linear Algebra and its Applications*.

Abraham A Ungar. 2005. *Analytic hyperbolic geometry: Mathematical foundations and applications*.

Shen Wang, Xiaokai Wei, Cicero Nogueira Nogueira dos Santos, Zhiguo Wang, Ramesh Nallapati, Andrew Arnold, Bing Xiang, Philip S Yu, and Isabel F Cruz. 2021. Mixed-curvature multi-relational graph neural network for knowledge graph completion. In *Proceedings of the Web Conference 2021*.

Song Wang, Xingbo Fu, Kaize Ding, Chen Chen, Huiyuan Chen, and Jundong Li. 2023. Federated few-shot learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

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*.

Bo Xiong, Shichao Zhu, Mojtaba Nayyeri, Chengjin Xu, Shirui Pan, Chuan Zhou, and Steffen Staab. 2022. Ultrahyperbolic knowledge graph embeddings. *ArXiv*.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th international conference on world wide web*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*.

Adnan Zeb, Summaya Saif, Junde Chen, James Jianqiao Yu, Qingshan Jiang, and Defu Zhang. 2024. Cope: Composition-based poincaré embeddings for link prediction in knowledge graphs. *Information Sciences*, 662:120197.

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

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

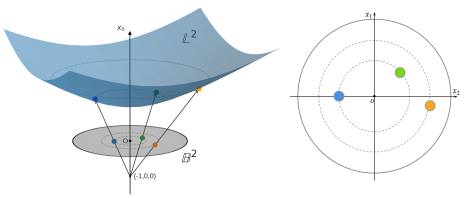
Yuying Zhao, Minghua Xu, Huiyuan Chen, Yuzhong Chen, Yiwei Cai, Rashidul Islam, Yu Wang, and Tyler Derr. 2024. Can one embedding fit all? a multi-interest learning paradigm towards improving user interest diversity fairness. In *Proceedings of the ACM on Web Conference*.

A Notations

Notation	Description
$\mathcal{G}$	A knowledge graph
$\mathcal{E}$	A set of entities
$\mathcal{R}$	A set of relations
h	Head entity
t	Tail entity
r	Relation
(h, r, t)	Fact in KG
$(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$	Embedding of a fact
$\boldsymbol{M}_.$	Mapping matrix
$f_{\boldsymbol{r}}(\boldsymbol{h}, \boldsymbol{t})$	Score function
$d_.(\cdot, \cdot)$	Distance metric in specific space
$\mathbb{R}^k$	k-dimensional Euclidean space
$\mathbb{C}^k$	k-dimensional complex space
$\mathbb{B}^k$	k-dimensional Poincaré ball model
$\mathbb{L}^k$	k-dimensional Lorentz model
$\circ$	Element-wise product
$\otimes$	Möbius matrix-vector multiplication
$\oplus$	Möbius addition
$\mathbf{O}(n)$	Orthogonal group
$\mathbf{O}(1, n)$	Lorentz group
$\mathbf{O}^+(1, n)$	Positive Lorentz group
$\mathbf{\Lambda}$	Lorentz transformations

Table A1: Notations

B More on Hyperbolic Geometry



(a) (b)

Figure A1: (a) Isometric transformation between the Poincaré ball model $\mathbb{B}^2$ (grey disk) and the Lorentz model $\mathbb{L}^2$ (blue surface); (b) Points on $\mathbb{B}^2$.

The Poincaré ball model. The k-dimensional Poincaré ball model of hperbolic space $\mathbb{B}^k$ is the interior of a unit ball in $\mathbb{R}^k$, i.e.,

$$\mathbb{B}^k = \{\boldsymbol{x} = [x_1, \ldots, x_k] \in \mathbb{R}^k : \|\boldsymbol{x}\| < 1\} \quad \text{(A1)}$$

where $\|\cdot\|$ is the Euclidean norm.

The correspondence between the Poincaré ball model $\mathbb{B}^k$ and the Lorentz model $\mathbb{L}^k$ is given by

$$[x_0, x_1, \ldots, x_k] \in \mathbb{L}^k \Leftrightarrow \left[\frac{x_1}{1+x_0}, \ldots \frac{x_k}{1+x_0}\right] \in \mathbb{B}^k, \quad \text{(A2)}$$

the stereographic projection of hyperbolic space (Cannon et al., 1997), as illustrated in Fig. A1 for the 2-dimensional case.

Geodesic distance. Recall that the geodesic distance in hyperbolic space is given by:

$$d_{\mathbb{L}}(\boldsymbol{x}, \boldsymbol{y}) = \cosh^{-1}(-\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbb{L}}) \triangleq \theta \quad \text{(A3)}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{L}^k$ and $\langle \cdot, \cdot \rangle_{\mathbb{L}}$ is the Lorentzian inner product.

Tangent space. Given $\boldsymbol{x} \in \mathbb{L}^k$, the *tangent space* centered at $\boldsymbol{x}$ is

$$T_{\boldsymbol{x}}\mathbb{L}^k := \{\boldsymbol{v} \in \mathbb{R}^{k+1} : \langle \boldsymbol{v}, \boldsymbol{x} \rangle_{\mathbb{L}} = 0\}. \quad \text{(A4)}$$

$T_{\boldsymbol{x}}\mathbb{L}^k$ is a local approximation of the Lorentz model at $\boldsymbol{x}$ and a Euclidean subspace of $\mathbb{R}^{k+1}$. $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathbb{L}^k$ is known as the *tangent vector*.

Exponential and logarithmic maps. The mapping between tangent space and hyperbolic space is conducted by *exponential and logarithmic maps*. For any $\boldsymbol{x} \in \mathbb{L}^n$ and $\boldsymbol{v} \in T_{\boldsymbol{x}}\mathbb{L}^n$, the exponential map at $\boldsymbol{x}$ is given by

$$\text{Exp}_{\boldsymbol{x}}(\boldsymbol{v}) = \cosh(\|\boldsymbol{v}\|_L)\boldsymbol{x} + \sinh(\|\boldsymbol{v}\|_L)\boldsymbol{v}/\|\boldsymbol{v}\|_L. \quad \text{(A5)}$$

the inverse of the exponential map, the logarithmic map, is given by

$$\text{Log}_{\boldsymbol{x}}(\boldsymbol{y}) = \frac{\theta}{\sinh(\theta)}(\boldsymbol{y} - \cosh(\theta)\boldsymbol{x}) \quad \text{(A6)}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{L}^n$ and θ is the geodesic distance between $\boldsymbol{x}$ and $\boldsymbol{y}$.

C Definitions

The definitions of relation patterns are presented in this section. With a slight abuse of notation, we denote a fact in the knowledge graph as $r(x, y)$, where $r \in \mathcal{R}$ is relation and $x, y \in \mathcal{E}$ are entities.

Definition 1. *A relation r is **symmetric** if* $\forall x, y$

$$r(x, y) \Rightarrow r(y, x) \quad \text{(A7)}$$

*A clause with such form is a **symmetric** pattern.*

Definition 2. *A relation r is **antisymmetric** if* $\forall x, y$

$$r(x, y) \Rightarrow \neg r(y, x) \quad \text{(A8)}$$

*A clause with such form is a **antisymmetric** pattern.*

Model	Score Function $f_{\boldsymbol{r}}(\boldsymbol{h},\boldsymbol{t})$	Embedding Spaces
SE (Bordes et al., 2011)	$-\|\boldsymbol{M}_{\boldsymbol{r},1}\boldsymbol{h}-\boldsymbol{M}_{\boldsymbol{r},2}\boldsymbol{t}\|_1^2$	$\boldsymbol{h},\boldsymbol{t}\in\mathbb{R}^k,\boldsymbol{M}_{\boldsymbol{r},i}\in\mathbb{R}^{k\times k}$
TransE (Bordes et al., 2013)	$-\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|_{1/2}$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{R}^k$
StransE (Nguyen et al., 2016)	$-\|\boldsymbol{M}_{\boldsymbol{r},1}\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{M}_{\boldsymbol{r},2}\boldsymbol{t}\|_{1/2}$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{R}^k,\boldsymbol{M}_{\boldsymbol{r},i}\in\mathbb{R}^{k\times k}$
RotatE (Sun et al., 2019)	$-\|\boldsymbol{h}\circ\boldsymbol{r}-\boldsymbol{t}\|_2$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{C}^k,\|\boldsymbol{r}_i\|=1$
RESCAL (Nickel et al., 2011)	$\langle\boldsymbol{h}^T\boldsymbol{M}_r\boldsymbol{t}\rangle$	$\boldsymbol{h},\boldsymbol{t}\in\mathbb{R}^k,\boldsymbol{M}_r\in\mathbb{R}^{k\times k}$
DistMult (Yang et al., 2015)	$\langle\boldsymbol{h}^T\mathrm{diag}(\boldsymbol{r})\boldsymbol{t}\rangle$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\mathrm{Re}(\langle\boldsymbol{h}^T\mathrm{diag}(\boldsymbol{r})\bar{\boldsymbol{t}}\rangle)$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{C}^k$
HAKE (Zhang et al., 2020)	$-\|\boldsymbol{h}_m\circ\boldsymbol{r}_m-\boldsymbol{t}_m\|_2$	$\boldsymbol{h}_m,\boldsymbol{t}_m\in\mathbb{R}^k,\boldsymbol{r}_m\in\mathbb{R}_+^k$
	$-\lambda\|\sin(\boldsymbol{h}_p+\boldsymbol{r}_p-\boldsymbol{t}_p)/2\|_1$	$\boldsymbol{h}_p,\boldsymbol{r}_p,\boldsymbol{t}_p\in[0,2\pi)^k,\lambda\in\mathbb{R}$
MuRP (Balazevic et al., 2019)	$-d_{\mathbb{B}}(\boldsymbol{M}_{\boldsymbol{r}}\otimes\boldsymbol{h},\boldsymbol{t}\oplus\boldsymbol{r}))^2+b_{\boldsymbol{h}}+b_{\boldsymbol{t}}$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{B}^k,\boldsymbol{M}_{\boldsymbol{r}}\in\mathbb{R}^{k\times k}$
AttH (Chami et al., 2019)	$-d_{\mathbb{B}}(Q(\boldsymbol{h},\boldsymbol{r}),\boldsymbol{t})^2+b_{\boldsymbol{h}}+b_{\boldsymbol{t}}$	$\boldsymbol{h},\boldsymbol{r},\boldsymbol{t}\in\mathbb{B}^k$
HyboNet (Chen et al., 2022b)	$-d_{\mathbb{L}}^2(g_{\boldsymbol{r}}(\boldsymbol{h}),\boldsymbol{t})+b_{\boldsymbol{h}}+b_{\boldsymbol{t}}+\delta$	$\boldsymbol{h},\boldsymbol{t}\in\mathbb{L}^k$
LorentzKG	$-d_{\mathbb{L}}^2(\boldsymbol{\Lambda}_{\boldsymbol{r},1}\boldsymbol{h},\boldsymbol{\Lambda}_{\boldsymbol{r},2}\boldsymbol{t})+b_{\boldsymbol{h}}+b_{\boldsymbol{t}}+\delta$	$\boldsymbol{h},\boldsymbol{t}\in\mathbb{L}^k,\boldsymbol{\Lambda}_{\boldsymbol{r},i}\in\mathbf{O}^+(1,n)$

Table A2: The score functions $f_{\boldsymbol{r}}(\boldsymbol{h},\boldsymbol{t})$ and embedding spaces of several knowledge graph embedding models, where $\circ$ denotes the Hadamard product, $\langle\cdot\rangle$ denotes the dot product, $\oplus$ and $\otimes$ denotes the Möbius addition and the Möbius matrix multiplication, respectively. Also, $d_{\mathcal{M}}(\cdot,\cdot)$ is the distance metric in manifold $\mathcal{M}$ and $b_{\boldsymbol{h}},b_{\boldsymbol{t}}\in\mathbb{R}$ are scalar biases. $\delta\in\mathbb{R}$ is a margin hyper-parameter. $Q(\boldsymbol{h},\boldsymbol{r})$ in AttH is a hyperbolic attention module, and $g_{\boldsymbol{r}}(\boldsymbol{h})$ in HyboNet is Euclidean linear transformation followed by normalization.

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

$$r_2(x,y)\Rightarrow r_1(y,x) \tag{A9}$$

*A clause with such form is a **inversion** pattern.*

Definition 4. *Relation r_1 is **composed** of relation r_2 and relation r_3 if $\forall x,y,z$*

$$r_2(x,y)\wedge r_3(y,z)\Rightarrow r_1(x,z) \tag{A10}$$

*A clause with such form is a **composition** pattern.*

D Proofs of Capability of LorentzKG in Modeling Relation Patterns

If a triple (h,r,t) is a valid fact, we would expect that in our embedding $\boldsymbol{\Lambda}_{r,1}\boldsymbol{h}=\boldsymbol{\Lambda}_{r,2}\boldsymbol{t}$, or equivalently $\boldsymbol{\Lambda}_{r,2}^{-1}\boldsymbol{\Lambda}_{r,1}\boldsymbol{h}=\boldsymbol{t}$. Note that $\boldsymbol{\Lambda}_{r,2}^{-1}\boldsymbol{\Lambda}_{r,1}\in\mathbf{O}^+(1,n)$ since $\mathbf{O}^+(1,n)$ is a group. In the following proofs of relation patterns, we denote $\boldsymbol{\Lambda}_{r,t}^{-1}\boldsymbol{\Lambda}_{r,1}$ as $\boldsymbol{\Lambda}_r$ and check if $\boldsymbol{\Lambda}_r\boldsymbol{h}=\boldsymbol{t}$ in a valid fact (h,r,t).

Proof. (symmetry pattern) If $r(x,y)$ and $r(y,x)$ hold, we have

$$(\boldsymbol{\Lambda}_r\boldsymbol{x}=\boldsymbol{y})\wedge(\boldsymbol{\Lambda}_r\boldsymbol{y}=\boldsymbol{x})\Rightarrow\boldsymbol{\Lambda}_r\boldsymbol{\Lambda}_r=I \tag{A11}$$

□

Proof. (antisymmetry pattern) If $r(x,y)$ and $\neg r(y,x)$ hold, we have

$$(\boldsymbol{\Lambda}_r\boldsymbol{x}=\boldsymbol{y})\wedge(\boldsymbol{\Lambda}_r\boldsymbol{y}\neq\boldsymbol{x})\Rightarrow\boldsymbol{\Lambda}_r\boldsymbol{\Lambda}_r\neq I \tag{A12}$$

□

Proof. (inversion pattern) If $r_1(x,y)$ and $r_2(y,x)$ hold, we have

$$(\boldsymbol{\Lambda}_{r_1}\boldsymbol{x}=\boldsymbol{y})\wedge(\boldsymbol{\Lambda}_{r_2}\boldsymbol{y}=\boldsymbol{x})\Rightarrow\boldsymbol{\Lambda}_{r_1}=\boldsymbol{\Lambda}_{r_2}^{-1} \tag{A13}$$

□

Proof. (compositon pattern) If $r_2(x,y),r_3(y,z)$ and $r_1(x,z)$ hold, we have

$$\begin{aligned}(\boldsymbol{\Lambda}_{r_2}\boldsymbol{x}=\boldsymbol{y})\wedge(\boldsymbol{\Lambda}_{r_3}\boldsymbol{y}=\boldsymbol{z})\wedge(\boldsymbol{\Lambda}_{r_1}\boldsymbol{x}=\boldsymbol{z})\\ \Rightarrow\boldsymbol{\Lambda}_{r_3}\boldsymbol{\Lambda}_{r_2}=\boldsymbol{\Lambda}_{r_1}\end{aligned} \tag{A14}$$

□

E Pseudocode for LorentzKG

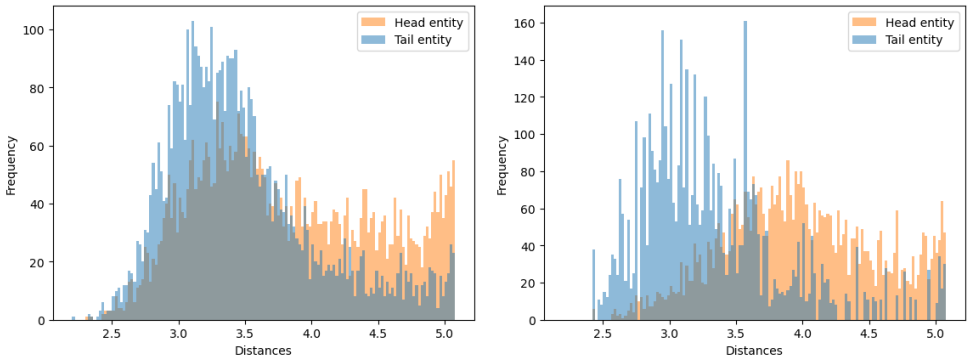
We illustrate the implementation of our proposed LorentzKG with pseudocode in Algorithm 1.

F Analysis on Hierarchy Pattern

In order to assess the hierarchy patterns captured by our LorentzKG model, we gather the embeddings of head and tail entities for some relations in WN18RR. Subsequently, we compute the hyperbolic distance from these embeddings to the origin point $\boldsymbol{o}=[1,0,\ldots,0]\in\mathbb{L}^k$, and plot the resulting distribution histograms, as illustrated in Fig. A2. We expect that entities at the same hierarchical level would exhibit similar distances from

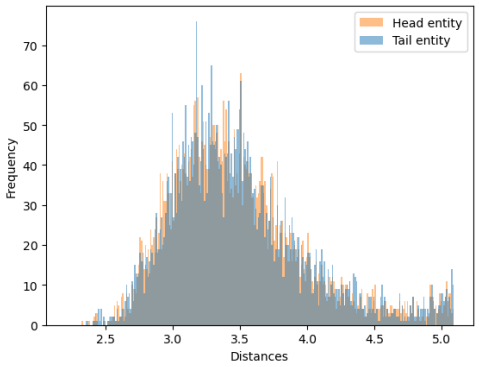
Algorithm 1: Pytorch-style Pseudocode for LorentzKG

```
# LorentzRotation: Lorentz rotation/reflection neural network module
# LorentzBoost: Lorentz boost neural network module
# LorentzDistance: squared Lorentzian distance function
#
# Train LorentzKG for N epochs
for epoch in range(N):
      for (h,r,t) in training data and negative samples:
         trans_h = LorentzRotation(h, r)
         trans_h = LorentzBoost(trans_h, r)
         trans_t = LorentzRotation(t, r)
         trans_t = LorentzBoost(trans_t, r)
         d = LorentzDistance(trans_h, trans_t) + bias_h + bias_t + margin
         loss = nn.BCEWithLogitsLoss(d, y) # y = ± 1
         loss.backward()
         optimizer_radam.step()
```

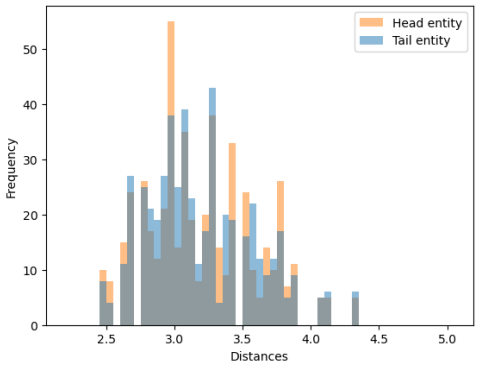


(a) _hypernym

(b) _instance_hypernym



(c) _derivationally_related_form



(d) _similar_to

Figure A2: Distribution histograms of entity embeddings in some relations in WN18RR. For a specific relation, the distribution of distances from the head entity embeddings to the origin point $\boldsymbol{o} = [1, 0, \ldots, 0] \in \mathbb{L}^k$ is shown in orange. And the distribution of distances from the tail entity embeddings to the origin point is shown in blue.

the origin point, while entities with higher-level hierarchies would be closer to the origin point.

In the WN18RR dataset, the relations *_hypernym* and *_instance_hypernym* are hierarchical relations where tail entities are at higher levels than head entities of the hierarchy. As depicted in Fig. 2(a) and Fig. 2(b), we observe that the tail embeddings tend to be situated at higher levels compared to the head embeddings, aligning with our expectations.

On the other hand, the relations *_derivation-*

	CoDEx-s		CoDEx-m		CoDEx-l	
Model	MRR	Hit@1	MRR	Hit@1	MRR	Hit@1
TransE	35.4	21.9	30.3	22.3	18.7	11.6
ComplEx	46.5	37.2	33.7	**26.2**	29.4	23.7
ConvE	44.4	34.3	31.8	23.9	30.3	24.0
TuckE	44.4	33.9	32.8	23.9	30.9	**24.4**
AttH	40.2	28.6	31.5	23.7	-	-
MuRP	42.0	31.1	30.6	22.6	-	-
CoPE	44.6	35.0	32.6	25.1	-	-
LorentzKG	**52.1**	**37.9**	**36.8**	24.5	**33.4**	22.6

Table A3: Performances of various KGE models on CoDEx datasets.

ally_related_form and *_similar_to* are symmetric relations. As shown in Fig. 2(c) and Fig. 2(d), the tail embeddings exhibit similar levels to the head embeddings, as anticipated.

G Additional Experimental Results

We validate our results on a new dataset, CoDEx (Safavi and Koutra, 2020). CoDEx, which comprises three diverse knowledge graphs of varying sizes (CoDEx-s, CoDEx-m, CoDEx-l), is a recently developed benchmark for KG completion tasks. The performance of our model, along with several competing models, is presented in the Table. A3 Most results are sourced from the original work(Safavi and Koutra, 2020), while the results from some hyperbolic KGE018 models (AttH and MuRP) are taken from a recent work, CoPE (Zeb et al., 2024). As evidenced in the table, our LorentzKG demonstrates a competitive to superior performance on the new benchmark.

H More on Model Capacity

Note that the number of parameters taken from the original GIE (Cao et al., 2022) paper for MuRP (Balazevic et al., 2019) is for an embedding dimension other than 32. For d=32, the number of parameters in MuRP is 5.41M (WN18RR) and 1.94M (FB15k-237), which is more than that in LorentzKG (1.39M and 1.42M). In LorentzKG, the number of parameters increases linearly with the number of entities and quadratically with the number of relations — the same complexity as in HyboNet (Chen et al., 2022b). However, since transformations are applied to both head and tail, whereas only the head is transformed in HyboNet, we have a few more parameters compared to HyboNet.

I Ethics Statement

This study adhered to rigorous ethical guidelines. All data used were publicly available and contained no personally identifiable information. No human participants or animals were involved, hence no ethical approval was required. The research was conducted responsibly, with any potential conflicts of interest declared. The findings are the result of unbiased scientific investigation.

J Use of AI Assistants

AI assistants are used in this paper for spelling and grammar check.