Multi-view Riemannian Manifolds Fusion Enhancement for Knowledge Graph Completion

Linyu Li, *Student Member, IEEE,* Zhi Jin†, *Fellow, IEEE,* Xuan Zhang, Haoran Duan, Jishu Wang, Zhengwei Tao, Haiyan Zhao, Xiaofeng Zhu, *Senior Member, IEEE*

***Abstract*—As the application of knowledge graphs becomes increasingly widespread, the issue of knowledge graph incompleteness has garnered significant attention. As a classical type of non-Euclidean spatial data, knowledge graphs possess various complex structural types. However, most current knowledge graph completion models are developed within a single space, which makes it challenging to capture the inherent knowledge information embedded in the entire knowledge graph. This limitation hinders the representation learning capability of the models. To address this issue, this paper focuses on how to better extend the representation learning from a single space to Riemannian manifolds, which are capable of representing more complex structures. We propose a new knowledge graph completion model called MRME-KGC, based on multi-view Riemannian Manifolds fusion to achieve this. Specifically, MRME-KGC simultaneously considers the fusion of four views: two hyperbolic Riemannian spaces with negative curvature, a Euclidean Riemannian space with zero curvature, and a spherical Riemannian space with positive curvature to enhance knowledge graph modeling. Additionally, this paper proposes a contrastive learning method for Riemannian spaces to mitigate the noise and representation issues arising from Multi-view Riemannian Manifolds Fusion. This paper presents extensive experiments on MRME-KGC across multiple datasets. The results consistently demonstrate that MRME-KGC significantly outperforms current state-of-the-art models, achieving highly competitive performance even with low-dimensional embeddings.**

***Index Terms*—Knowledge Graph, Knowledge Graph Completion, Contrastive Learning, Hyperbolic Space, Riemannian Manifolds.**

I. Introduction

THE emergence of knowledge graphs [1] [2] has significantly advanced fields such as knowledge engineering and artificial intelligence. This includes applications in recommendation systems [3] [4], knowledge-enhanced large language models [5] [6] [7], biomedicine [8] [9] [10], temporal data [11] and financial risk control [12] [13]. Unfortunately, even the most renowned large-scale open knowledge graphs, such as Wikidata [14] and ConceptNet [15], suffer from significant incompleteness. This greatly limits the performance of knowledge graphs in downstream applications.

This work was supported by the National Natural Science Foundation of China under Grant No.62436006.

Linyu Li, Zhi Jin, Zhengwei Tao, and Haiyan Zhao are with Key Laboratory of High Confidence Software Technologies (PKU), Ministry of Education, and School of Computer Science, Peking University, Beijing 100871, China (e-mail: xltx_youxiang@qq.com; zhijin@pku.edu.cn; tttzw@stu.pku.edu.cn; zhhy.sei@pku.edu.cn). Xuan Zhang and Jishu Wang are with School of Software, Yunnan Key Laboratory of Software Engineering, Yunnan University, Kunming 650091, China (e-mail: zhxuan@ynu.edu.cn; cswangjishu@hotmail.com). Haoran Duan is with School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: hrduan07@gmail.com). Xiaofeng Zhu is with University of Electronic Science and Technology of China, Chengdu, Sichuan 610056, China (e-mail: seanzhuxf@gmail.com).

Corresponding authors: Zhi Jin (zhijin@pku.edu.cn).

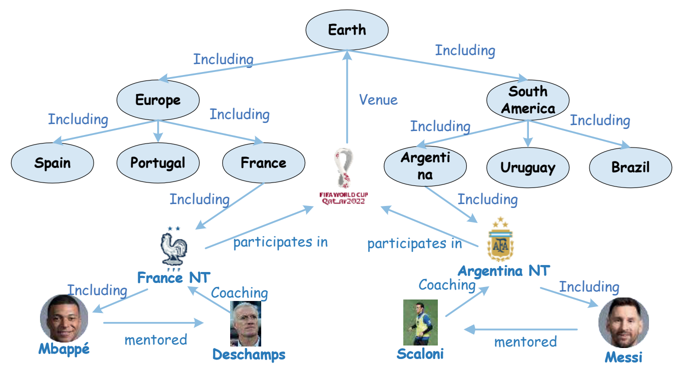


Fig. 1. An example illustrating how KGs simultaneously encompass hierarchical structures, chain-transitive structures, and cyclic structures.

A Knowledge Graph (KG) typically stores real-world knowledge in the form of triples, representing the factual semantic relationships between entities. In order to deal with the issue of incompleteness, researchers propose the task of Knowledge Graph Completion (KGC), which aims to infer missing facts or predict incomplete facts (triples) based on existing knowledge, thereby completing the KG. It can be a prediction of the tail entity given the head entity and the relation $(h, r, ?)$, or a prediction of the head entity given the relation and the tail entity $(?, r, t)$.

Using knowledge graph embeddings to infer missing facts is a forward-looking method, which aims to learn the embeddings of entities and relationships in a low-dimensional vector space, i.e., capturing the patterns of connections between facts (entities and relationships) by constructing scoring functions within the corresponding space. In simple terms, finding the most suitable embeddings and embedding space for knowledge graphs is crucial and highly challenging. For example, KGC models based on zero-curvature Euclidean space, such as TransE [16] and Mixup-enhanced embeddings [21], provide simple, yet effective solutions. Complex space-based methods such as RotatE [17] model intricate relations through rotation, while approaches using geometric operation combinations [18] [20] or multi-functional embedding spaces [19] enable more flexible relationship modeling. Non-Euclidean spaces also offer significant advantages. Hyperbolic embeddings, such as low-dimensional hyperbolic models [22] and relation-specific

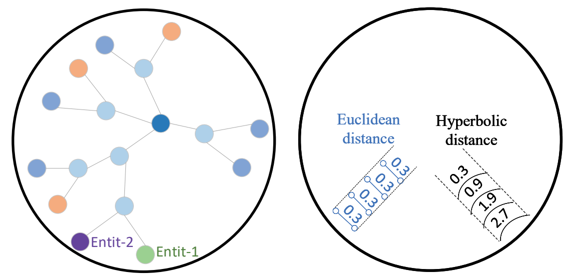
This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3538110

hyperbolic cones [23], capture hierarchies and long-range dependencies effectively. Models like normalizing flows [24] or mixed-curvature spaces [25] further improve flexibility and expressiveness.

Firstly, as a typical non-Euclidean knowledge structure, KGs require robust modeling capabilities to capture various complex structures, which many previous studies [16] [17] [22] have overlooked. As illustrated in Fig. 1, real-world knowledge often exhibits different patterns and complex geometric shapes (e.g., hierarchical structures, circular structures, linear transitive structures, etc.). Euclidean space and hyperbolic space, as special forms of Riemannian manifolds with constant curvature, are limited in simultaneously modeling the aforementioned complex structures of various KGs. Using single-curvature Riemannian manifolds to model KGs cannot adequately capture the diverse patterns and complex geometries shown in Fig. 1.

Secondly, hierarchical structures, being the most important structure of knowledge graphs, are well supported by hyperbolic space for KG modeling. In hyperbolic space, the embedding space expands exponentially with increasing embedding dimensions [26], whereas in Euclidean space it grows polynomially. This characteristic provides a solid foundation for embedding hierarchical structures supported by hyperbolic geometry. Consequently, due to the exponential growth of volume in hyperbolic space, it can capture the complexity of hierarchical data more efficiently, allowing it to embed more information than Euclidean space at the same dimensionality [74]. However, the current work in KGC [22] [27] mainly studies the Poincaré Ball Model, but lacks of the consideration of the Lorentz Model.

This paper argues that although the Poincare sphere model can effectively capture the semantic information embedded in the knowledge graph hierarchy, it also has limitations. This insufficiency stems from the inherent limitations of the Poincaré Ball Model, particularly in preserving hierarchical relationships as embeddings move closer to the boundary of the hyperbolic space, where the density of representations increases rapidly, potentially leading to distortions in semantic representation. This effect has been noted in prior works, such as [74] and [51], which highlight that the boundary-induced compression of points in hyperbolic space can hinder the effective capture of hierarchical information. Moreover, the exponential/logarithmic mappings in the Poincaré Ball Model present numerical stability issues (which can lead to “NaN” problems in practice) [28] [29]. Recent work in other fields has attempted to address this problem. For example, [29] uses the random feature mapping with the Laplace operator’s eigenfunctions and [30] uses the Lorentz-type operators to replace exponential/logarithmic mappings. However, the issue of numerical stability remains largely unresolved. This paper proposes that for the KGC task, rather than discarding the Poincaré Ball Model, we should combine it with the Lorentz Model to alleviate numerical instability, as the Poincaré Ball Model is effective in modeling hierarchical relationships but suffers from numerical instability and boundary effects, while the Lorentz Model [76] avoids these issues and offers better numerical stability, leveraging the strengths of both models to better capture the rich semantic information and hierarchical relationships within KGs. Subsequent experiments consistently demonstrate the validity of this perspective.



(a) Example of hyperbolic modeling.

(b) Hyperbolic distance and Euclidean distance.

Fig. 2. Illustration of the marginal density explosion problem for the Poincare sphere hyperbolic model and its comparison with Euclidean space distance [26] .

Along this line, this paper proposes an expansion from a single Euclidean space and hyperbolic space to a model called MRME-KGC, which fuses views from Riemannian manifolds with zero curvature, positive curvature, and negative curvature. Specifically, MRME-KGC employs an attention mechanism to fuse two negative curvature Riemannian manifold views (Poincaré Ball Model and Lorentz Model) for more effectively capturing the embeddings of hierarchical structures in the KG, a zero curvature Riemannian manifold view (Euclidean space) for more effectively capturing the embeddings of linear transitive structures in the KG, and a positive curvature Riemannian manifold view (Spherical space) for more effectively capturing the embeddings of circular structures in the KG. However, there is one more issue worth noting. As shown in Fig. 2(b), in a hyperbolic space with constant negative curvature, the nodes near the center of the graph are closer together, while those near the boundary are farther apart. This can result in situations as depicted in Fig. 2(a), where the distance between two entities located at the edges of the hyperbolic space is very large, despite them having a potential semantic similarity in reality. To better address this issue and to reduce the noise when fusing hyperbolic space with spherical space, we propose a contrastive learning method for KGC in Riemannian manifolds. Furthermore, this paper proposes employing Lorentz linear transformations as the relation transformation function for head entities in the Lorentz Model.

The specific contributions of this paper are as follows.

- This paper points out that existing models for KGC tasks based on hyperbolic space only consider using the Poincaré Ball Model to represent hyperbolic space, neglecting the simultaneous use of both the Lorentz Model and the Poincaré Ball Model, which severely limits the model performance.
- To the best of our knowledge, this paper is the first to explore the use of a two-view hyperbolic representation model for combining zero-curvature and positive-curvature Riemannian manifolds for KGC tasks. Through experiments, we found that models integrating multiple views simultaneously can more accurately reflect hierar-

chical structures and achieve better performance.
- This paper proposes a contrastive learning method for KGC tasks under Riemannian manifolds, aiming to mitigate the potential noise generated during the fusion of the Poincaré Ball Model view with the spherical view and to reduce the distances between edge entities in the Poincaré Ball Model.
- Through extensive experiments on six datasets, we found that the MRME-KGC model not only achieves superior performance on most metrics but also effectively captures complex structural semantic information in knowledge graphs under low-dimensional conditions. Furthermore, compared to other models in standard dimensions, MRME-KGC significantly improves metrics such as Hits@1 and MRR, achieving optimal results.

The remainder of this paper is organized as follows. Section II introduces some related work on KGC and approaches based on Riemannian Manifolds. Section III presents the preliminaries and a detailed explanation of the MRME-KGC method. In Section IV, we conduct extensive experiments and analyses from various perspectives to demonstrate the effectiveness of our approach. Finally, Section V concludes the paper and provides an outlook on future work.

II. Related Work

Many advancements in KGC have greatly benefited from the exploration of non-Euclidean geometric properties and the precise modeling of more complex Riemannian Manifold spaces. In this section, we first introduce KGC models based on Euclidean space and complex space. Secondly, we introduce the KGC models based on Riemannian Manifolds. Finally, we discuss the applications of Riemannian Manifolds in other fields.

A. KGC Models Based on Euclidean and Complex Spaces

1) KGC model based on translation representation: KGC models based on Euclidean space are the most extensively researched type to date. TransE [16] is a typical representative of this category, which applies the concept of word embeddings combined with vector addition and subtraction operations to the embedding representation learning of knowledge graphs. To address the limitations of TransE in handling more complex types of relationships, subsequent models such as TransD [31], TransR [32] and TorusE [33] propose various improvements.

In subsequent work, researchers apply mathematical operations from Complex Space to the KGC task, including models such as ComplEx [34], RotatE [17], QIQE [35] and HAKE [36]. ComplEx is the first work to embed entities and relationships into complex space, capturing symmetric and antisymmetric relations through complex inner products. RotatE conceptualizes relationships as rotations in complex spaces and has become one of the most popular KGC models. Researchers extend the exploration beyond the single real and imaginary parts of complex space to hyper-complex spaces (quaternion spaces [37]). QuatE [38] embeds entities and relationships into quaternion space and captures complex relationships through quaternion multiplication.

RelEns-DSC [62] is a relation-aware ensemble learning approach for knowledge graph embedding that independently optimizes the ensemble weights for different relations using a divide-search-combine strategy, thus enhancing both efficiency and performance. CompoundE [20] introduces a novel knowledge graph embedding model that combines translation, rotation, and scaling operations in capturing diverse relational patterns. GreenKGC [18] is a lightweight and modular knowledge graph completion method that integrates representation learning, feature pruning, and decision learning, achieving competitive performance in low-dimensional settings while significantly reducing model size and improving applicability.

2) KGC Models Based on Neural Network: In the KGC task, neural network models have received widespread attention due to their powerful representation ability and flexibility. ConvE [40], ConvKB [41], CLGAT [39] and InteractE [42] are typical representatives of using neural networks for KGC. They capture the complex interactions between entities and relations through different convolution or graph convolution operations, each with its own characteristics. These models use various convolution operations to improve the effectiveness of knowledge graph completion. However, these methods of learning KG embedding in a data-driven manner still have difficulty capturing the heterogeneity of complex structures and relations in knowledge graphs.

B. KGC Models Based on Riemannian Manifolds

Models based on Riemannian Manifolds embed knowledge graphs into non-Euclidean spaces to effectively capture their inherent complex geometric structures. This approach helps to more accurately represent the associations between entities and relationships in knowledge graphs, thereby enhancing the model's ability for graph modeling and reasoning. The classic KGC models based on Riemannian Manifolds include MuRP [43], AttH [22], ConE [23], MuRMP [25] and GIE [27]. Among them, MuRP achieves effective modeling of hierarchical structures in knowledge graphs by embedding multiple types of relationships in hyperbolic space. AttH utilizes attention mechanisms to selectively focus on important relational features in hyperbolic space, enhancing the model's expressiveness and reasoning performance. ConE uses cone hyperbolic space embedding to further improve the ability to represent complex relations and hierarchical structures. These methods perform well in processing knowledge graph data with hierarchical structures and complex relations.

C. Related applications of Riemannian Manifolds

In recent years, Riemannian manifolds have seen significant advancements in their various applications within deep learning, particularly excelling in handling complex, non-Euclidean structured data. In the task of time series graphing, HTGN [44] embeds temporal networks into hyperbolic space. For recommendation systems, LKGR [45] proposes a knowledge-aware recommendation method based on a hyperbolic geometric Lorentz model. HGCF [46] introduces a hyperbolic GCN model for collaborative filtering. In natural language processing tasks, HyboNet [47] extends word embeddings

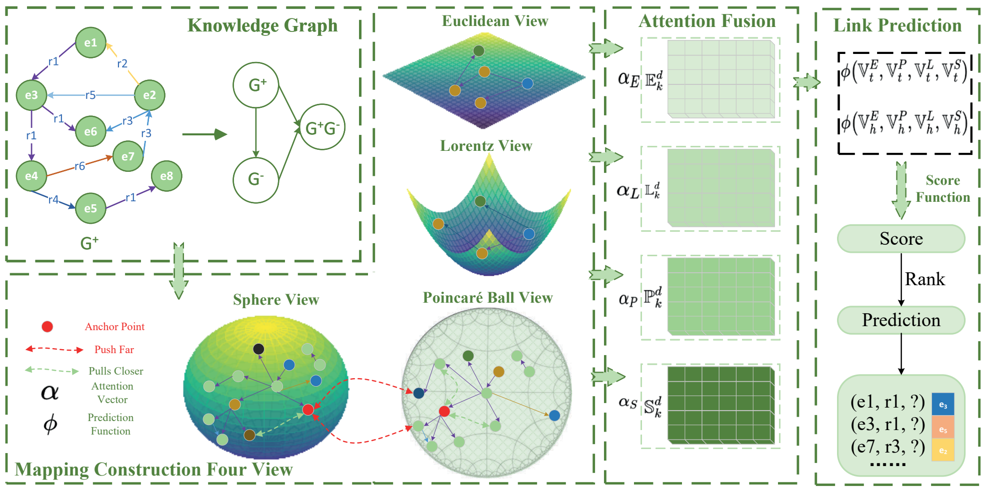


Fig. 3. Framework of MRME-KGC. The embeddings in MRME-KGC incorporate views from four distinct spaces, integrating multiple view embeddings through an attention mechanism. MRME-KGC employs the iterative entity representation updates and the contrastive learning to achieve representation learning. The optimized entity and relation representations are then fed into the scoring function to perform relevant triple link prediction.

TABLE I
NOTATION SUMMARY

Notation	Explanation
$\mathcal{M}$	Riemannian manifold
$\mathcal{T}_x\mathcal{M}$	Riemannian manifold tangent space
$\mathcal{G}$	KG Set
$\mathcal{E}$	Entity Set
$\mathcal{R}$	Relation Set
$\mathcal{T}$	Triple Set
$f(\cdot)$	Score Function
$exp^k_{\boldsymbol{x}(\boldsymbol{\alpha})}$	Exponential Map
$log^k_{\boldsymbol{x}(\boldsymbol{\alpha})}$	Logarithmic Map
$\mathbb{E}^d_k$	Euclidean Space
$\mathbb{P}^d_k$	Poincaré Ball Model Space
$\mathbb{L}^d_k$	Lorentz Model Space
$\mathbb{S}^d_k$	Sphere Space
τ	Temperature Parameters
$g_{\boldsymbol{r}}()$	Relational Conversion Functions
ξ_{G_r}	Curvature Estimate
Khs_{G_r}	Krackhardt Hierarchy Score

from traditional Euclidean space to Riemannian Manifolds using Fully Hyperbolic Neural Networks, achieving notable results in machine translation tasks. Additionally, Riemannian Manifolds are extensively used in deep learning-based graph embedding tasks, including applications such as HypDiff [48], HyGCAT [77] and MotifRGC [28].

III. Methodology

In this section, we first introduce the prerequisites to define the Riemannian Manifolds space. Then we introduce the MRME-KGC model in detail.

A. Preliminaries and Problem Formulation

1) Riemannian manifold: A Riemannian manifold $\mathcal{M}$ can be concisely defined as a smooth manifold coupled with a Riemannian metric. In geometric mathematics, most manifolds can be regarded as Riemannian manifolds $(\mathcal{M}, g_x)$ of dimension n. At every point $x \in \mathcal{M}$ within the Riemannian manifold, the Riemannian metric g_x is defined on its tangent space $\mathcal{T}_x\mathcal{M}$. Typically, the conversion between manifold vectors and their tangent space counterparts is achieved through the logarithmic and exponential maps. Specifically, $\log_x : \mathcal{M} \to \mathcal{T}_x\mathcal{M}$ denotes the conversion of a vector in the Riemannian metric to its tangent space, while $\exp_x : \mathcal{T}_x\mathcal{M} \to \mathcal{M}$ represents the inverse conversion via the exponential mapping. For more detailed definitions, please refer to [26] [49] [50].

2) Curvature: The curvature K is a measure of the flatness of the surface and determines the shape of the Riemannian manifold. Each point x in the Riemannian manifold is associated with a curvature k and a corresponding radius of curvature $\frac{1}{|\kappa_x|}$. Specifically, k determines the curvature of space, and there are three different types of $\mathcal{M}$ depending on the curvature. If the constant curvature is negative, the manifold is a hyperbolic space $\mathbb{H}_K$; if the curvature is positive, the manifold is a spherical space $\mathbb{S}_K$; Euclidean space $\mathbb{E}_K$ is considered a typical example of a manifold with zero curvature. Due to the differences in operations within hyperbolic and spherical spaces compared to Euclidean space, we need to utilize logarithmic and exponential maps to facilitate the conversion between each space and its respective tangent space.

3) Problem Formulation: First, this paper defines a KG as: $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ and $\mathcal{R}$ represent the sets of entities and relations in the KG, respectively. $\mathcal{T} = \{(e_h, r, e_t) \mid e_h, e_t \in \mathcal{E}, r \in \mathcal{R}\}$, which represents a set of triple combinations. Then, the KGC is formalized as a link prediction task. It can be $(h, r, ?)$ or $(?, r, t)$, i.e., given relation r of a triplet and head entity h or tail entity t, to predict tail entity t or head entity h. By learning the vector representation of entity $\mathcal{E}$ and relation $\mathcal{R}$, the (h, r, t) triple is represented by a $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$ vector, where $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{V}_e, \mathbf{r} \in \mathbb{V}_r$ ($\mathbb{V}$ represents a vector space of d-dimensions, d is the vector dimension parameter). For example, $\mathbb{V} = \mathbb{E}_k$ in TransE [16] (Euclidean space), $\mathbb{V} = \mathbb{C}_k$ in RotatE [17] (complex space), and $\mathbb{V} = \mathbb{H}_k$ in RttH [22] (hyperbolic space). Additionally, the notations used throughout this paper are summarized in Table I.

To more effectively learn the various complex structures within a KG and to mitigate the issue of overly dense boundary distributions in the Poincaré Ball Model embeddings, as well as to reduce the noise generated during the fusion of the Poincaré Ball Model and spherical spaces, this study proposes the MRME-KGC model. The architecture of the model is depicted in Fig. 3. Unlike previous works [16] [22] that defined the embedding space in either Euclidean space $\mathbb{E}_k$ or hyperbolic space $\mathbb{H}_k$, the MRME-KGC model incorporates multiple views of KG representations. These views include the Poincaré Ball view $\mathbb{P}_k$, Lorentz view $\mathbb{L}_k$, Sphere view $\mathbb{S}_k$, and Euclidean space view $\mathbb{E}_k$. The model combines relations and entities through a translation function. In addition, MRME-KGC enhances the representation learning of Poincaré Ball and sphere space through contrastive learning. Finally, the semantic information contained in the complex structure of KG is captured by combining spatial feature fusion with attention mechanism. The MRME-KGC model predicts the tail entity using a specific relation translation function $g_r : \mathbb{V}_{e_1} \to \mathbb{V}_{e_2}$, this function translates the head entity to the tail entity, i.e., $g_r(\mathbf{h}) = \mathbf{t}$, and is used to measure the distance between e_1 and e_2. Based on this translation function, we define the scoring function as follows:

$$f : \mathbb{V}_{e_1} \times \mathbb{V}_r \times \mathbb{V}_{e_2} \to \text{score} \in \mathbb{R} \tag{1}$$

The score is used to measure the accuracy of each triple. Generally speaking, the score function can be formalized as: $f(\mathbf{h}, \mathbf{r}, \mathbf{t}) = p\left(g_r(\mathbf{h}), \mathbf{t}\right)$, where $f(\mathbf{h}, \mathbf{r}, \mathbf{t})$ evaluates the plausibility of the triplet (h, r, t) by computing the distance between the translated head entity and the tail entity in the embedding space.

B. Detailed Description of MRME-KGC

1) Riemannian Manifolds Representation Learning: We need to obtain representations of entities in different view spaces. Euclidean space, hyperbolic space, and spherical space have different properties, and their spatial structures are established through different metrics (such as the Euclidean metric or hyperbolic metric). To this end, this paper uses the mapping operation proposed in [51] to project the d-dimensional points in the Euclidean space $\left(\mathbb{E}_K^d, g^{\mathbb{E}}\right)$ into the Poincaré ball $\left(\mathbb{P}_{-k}^d, g^{\mathbb{P}}\right)$ with a curvature of $-k$ $(-k < 0)$ and the sphere space $\left(\mathbb{S}_K^d, g^{\mathbb{S}}\right)$ with a curvature of k $(k > 0)$ through $\exp_x : \mathcal{T}_x\mathcal{M} \to \mathcal{M}$, and then project them back to the Euclidean space using the $\log_x : \mathcal{M} \to \mathcal{T}_x\mathcal{M}$ operation. For each point v in the tangent space $\mathcal{T}_x\mathcal{M}$ (Euclidean space), and the points x and y in the Riemannian space, they can be converted using the following formulas (2) and (3).

$$\exp_{\boldsymbol{x}}^k(\boldsymbol{v}) = \boldsymbol{x} \oplus_K \left(\tanh\left(\sqrt{k}\frac{\lambda_{\boldsymbol{x}}\|\boldsymbol{v}\|}{2}\right)\frac{\boldsymbol{v}}{\sqrt{k}\|\boldsymbol{v}\|}\right) \tag{2}$$

$$\log_{\boldsymbol{x}}^k(\boldsymbol{y}) = \frac{2}{\sqrt{k}\lambda_{\boldsymbol{x}}}\tanh^{-1}\left(\sqrt{k}\,\|-\boldsymbol{x} \oplus_k \boldsymbol{y}\|\right)\frac{-\boldsymbol{x} \oplus_k \boldsymbol{y}}{\|-\boldsymbol{x} \oplus_k \boldsymbol{y}\|} \tag{3}$$

$$\boldsymbol{x} \oplus_k \boldsymbol{y} = \frac{\left(1 + 2k\langle\boldsymbol{x}, \boldsymbol{y}\rangle + k\|\boldsymbol{y}\|^2\right)\boldsymbol{x} + \left(1 - k\|\boldsymbol{x}\|^2\right)\boldsymbol{y}}{1 + 2k\langle\boldsymbol{x}, \boldsymbol{y}\rangle + k^2\|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2} \tag{4}$$

where $\lambda_{\boldsymbol{x}} = \frac{2}{1-k\|\boldsymbol{x}\|^2}$ is the Riemannian metric,$\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathcal{M}_k^n$, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{M}_k^d$. $\oplus_c$ represents the Möbius addition operation proposed in [52]. Due to transformations in the overall space, many operations that are common in Euclidean space are no longer applicable in other spaces. In the KGC task, the most crucial aspect is measuring the distance between entities. Distance measurement in deep learning often involves operations such as addition, multiplication and inner products. Fortunately, previous work [51] has extended the relevant distance formulas from Euclidean space to Riemannian manifolds through tangent space mapping. As a result, the distance formula between any two points u and v in Poincaré ball and sphere space can be defined as:

$$d_k(\boldsymbol{u}, \boldsymbol{v}) = \frac{2}{\sqrt{k}}\tanh^{-1}\left(\sqrt{k}\,\|-\boldsymbol{u} \oplus_c \boldsymbol{v}\|\right) \tag{5}$$

Consequently, we obtain entity representations in the d-dimensional Euclidean space, the Poincaré ball space, and the spherical space, denoted as $\mathbb{E}_k^d(k = 0)$, $\mathbb{P}_k^d(k < 0)$, and $\mathbb{S}_k^d(k > 0)$, respectively. To alleviate potential numerical instability problems in the Poincaré ball model [28] [29] and to better capture hierarchical semantic relationships, MRME-KGC employs both the Poincaré ball and Lorentz hyperbolic models. This paper formalizes the n-dimensional Lorentz model as a Riemannian manifold $\left(\mathcal{L}^d, \mathfrak{g}_{x^L}\right)$, where $\mathfrak{g}_{\boldsymbol{x}^L} = \operatorname{diag}(-1, 1, \cdots, 1)$is the Riemannian metric. $\mathbb{L}_K^n = (\mathcal{L}^n, \mathfrak{g}_{x^K})$ satisfies $\mathcal{L}^d := \left\{\boldsymbol{x} \in \mathbb{R}^{d+1} \,\middle|\, \langle\boldsymbol{x}, \boldsymbol{x}\rangle_{\mathcal{L}} = \frac{1}{K}, x_a > 0\right\}$, where $\langle,\rangle_{\mathcal{L}}$ represents the Lorentz inner product. The Lorentz inner product of two points x, y is defined as:

$$\begin{aligned} \langle\boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}} &:= -x_a y_a + \boldsymbol{x}_b^\top \boldsymbol{y}_b \\ &= \boldsymbol{x}^\top \operatorname{diag}(-1, 1, \cdots, 1)\boldsymbol{y} \end{aligned} \tag{6}$$

Given every point x in $\mathbb{L}_K^n$, there exists the following form: $\boldsymbol{x} = \begin{bmatrix} x_a \\ \boldsymbol{x}_b \end{bmatrix}, \boldsymbol{x} \in \mathbb{R}^{n+1}, x_a \in \mathbb{R}, \boldsymbol{x}_b \in \mathbb{R}^n$ For simplicity, in the following part of this paper, the points belonging to the Lorentz model are formalized as $\boldsymbol{x} \in \mathbb{L}_K^d$. The tangent space for a given $\mathbb{L}_K^d$ at x is defined as:

$$\mathcal{T}_{\boldsymbol{x}}\mathbb{L}_K^d := \left\{\boldsymbol{y} \in \mathbb{R}^{d+1} \mid \langle\boldsymbol{y}, \boldsymbol{x}\rangle_{\mathcal{L}} = 0\right\} \tag{7}$$

where $\mathcal{T}_{\boldsymbol{x}}\mathbb{L}_K^d$ means that $\mathcal{T}_{\boldsymbol{x}}\mathbb{L}_K^d$ is in the orthogonal space of the Lorentzian inner product of the space where point x is located. Unlike the exponential/logarithmic mapping of the Poincaré ball, the logarithmic and exponential mapping of the Lorentzian model is defined as:

$$\begin{aligned} &\exp_{\boldsymbol{x}(\boldsymbol{\alpha})}^k = \cosh(\sqrt{-k}\|\boldsymbol{\alpha}\|_{\mathcal{L}})\boldsymbol{x} + \sinh(\sqrt{-k}\|\boldsymbol{\alpha}\|_{\mathcal{L}})\frac{\boldsymbol{\alpha}}{\sqrt{-k}\|\boldsymbol{\alpha}\|_{\mathcal{L}}} \\ &\|\boldsymbol{\alpha}\|_{\mathcal{L}} = \sqrt{\langle\boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle_{\mathcal{L}}}. \end{aligned} \tag{8}$$

$$\begin{aligned} \log_{\boldsymbol{x}(\boldsymbol{y})} &= \frac{\cosh^{-1}(\beta)}{\sqrt{\beta^2 - 1}}(\boldsymbol{y} - \beta\boldsymbol{x}), \\ \beta &= k\langle\boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}} \end{aligned} \tag{9}$$

the logarithmic map $\exp_{\boldsymbol{x}}^k(\boldsymbol{\alpha}) : \mathcal{T}_{\boldsymbol{x}}\mathbb{L}_k^d \to \mathbb{L}_k^d$ maps any tangent vector $\boldsymbol{\alpha} \in \mathcal{T}_{\boldsymbol{x}}\mathbb{L}_k^d$ to $\mathbb{L}_k^d$. The logarithmic mapping $\log_{\boldsymbol{x}}{}^k(\boldsymbol{y}) : \mathbb{L}_k^d \to \mathcal{T}_{\boldsymbol{x}}\mathbb{L}_k^d$ is used to map any vector $\boldsymbol{y} \in \mathbb{L}_K^n$ to the tangent

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3538110

vector $\mathcal{T}_{\boldsymbol{x}}\mathbb{L}_k^d$, which plays the opposite role of the exponential mapping.

2) Multi-view Riemannian Manifolds Fusion Enhancement: In this section, we first generate embeddings of the four spatial views for entity e and relation r in the KG, including embedding $\mathbb{E}_k^d$ in the Euclidean space, embedding $\mathbb{P}_k^d$ of the Poincare disk in the hyperbolic space, embedding $\mathbb{L}_k^d$ of the Lorentz model, and embedding $\mathbb{S}_k^d$ in the spherical space. Then we obtain the head entity representation and the tail entity representation of the four views after the relation conversion function $g_r(\boldsymbol{e})$.

For transformed head entity $\mathbb{V}_h^E$ in the Euclidean space, we initialize h and r and obtain it through the relational transformation function $g_r^{\mathbb{E}}()$ in the Euclidean space:

$$\mathbb{V}_h^E = g_r^{\mathbb{E}}(\mathbf{h}) = r \cdot h \tag{10}$$

where $\mathbb{V}_h^E \in \mathbb{E}_k^d$, $g_r^{\mathbb{E}}(\mathbf{h})$ represents the transfer function in Euclidean space. Similar to the conversion function of Euclidean space, we can use the conversion functions $g_r^{\mathbb{P}}$ and $g_r^{\mathbb{S}}$ to obtain the head entity $\mathbb{V}_h^P$ in the Poincaré ball model and the head entity $\mathbb{V}_h^S$ in the Sphere Space respectively:

$$\mathbb{V}_h^P = g_r^{\mathbb{P}}(\mathbf{h}) = r \otimes_k \exp_{\mathbb{P}}^k(h) \tag{11}$$

$$\mathbb{V}_h^S = g_r^{\mathbb{S}}(\mathbf{h}) = r \otimes_k \exp_{\mathbb{S}}^k(h) \tag{12}$$

$$\boldsymbol{r} \otimes_k \boldsymbol{x} = \frac{1}{\sqrt{k}} \tanh\left(r \tanh^{-1}(\sqrt{k}\|\boldsymbol{x}\|)\right) \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} \tag{13}$$

where $\mathbb{V}_h^P \in \mathbb{P}_k^d$,$\mathbb{V}_h^S \in \mathbb{S}_k^d$,$\boldsymbol{r} \in \mathbb{R}$,$\otimes_k$ represent the Möbius multiplication operation.

Inspired by the applications of the Lorentz model in other fields [53] [47] [54], we define the transfer function in a different way from the Poincaré ball model. Consistent with [53], we first initialize the Lorentz space parameters using a Gaussian distribution in the tangent space, use an exponential mapping to map the embedding to the hyperbolic space to obtain the head entity $\boldsymbol{h}$, and then use the Lorentz linear transformation to define the relation transformation function $g_r^{\mathbb{L}}(\mathbf{h})$:

$$g_{\boldsymbol{r}}^L\left(\begin{bmatrix} \boldsymbol{h}^\top \\ \boldsymbol{W} \end{bmatrix}\right) = \begin{bmatrix} \frac{\sqrt{|\mathbf{Wr}|-1/k}}{\mathbf{h}\,\mathbf{r}}\mathbf{h}^\top \\ \mathbf{W} \end{bmatrix} \tag{14}$$

where $\boldsymbol{h} \in \mathbb{L}_k^{d+1}, \boldsymbol{W} \in \mathbb{R}^{d\times(d+1)}$, $g_r^{\mathrm{L}}(\mathbf{h}) \in \mathbb{R}^{(d+1)\times(d+1)}$ is the Lorentz linear transformation of the relation $\boldsymbol{r}$ on the head entity through the learnable weight matrix $\mathbf{W}$. The aim is to learn a function that maps any matrix to a matrix suitable for hyperbolic linear layers. Additionally, to ensure consistency in the view dimensions across the four spaces, we define the entities obtained from the Lorentz model, after applying the relation transformation function, as follows:

$$\mathbb{V}_h^{\mathbb{L}} = g_r^{\mathbb{L}}\left(\begin{bmatrix} \boldsymbol{h}^\top \\ \boldsymbol{W} \end{bmatrix}\right) r \tag{15}$$

where $\mathbb{V}_h^{\mathrm{L}} \in \mathbb{L}_K^d$ represents the head entity representation of the Lorentz model after the transformation function, we use Theorem 1. to prove the rationality of the dimensional transformation.

Theorem 1. For entities and relationships in the Lorentz model, there can be changes in the following dimensions:

$$\forall \boldsymbol{r} \in \mathbb{L}_k^d, \forall \begin{bmatrix} \boldsymbol{h}^\top \\ \boldsymbol{W} \end{bmatrix} \in \mathbb{R}^{(d+1)\times(d+1)} \Rightarrow g_{\boldsymbol{r}}^L(\begin{bmatrix} \boldsymbol{h}^\top \\ \boldsymbol{W} \end{bmatrix})\boldsymbol{r} \in \mathbb{L}_k^d \tag{16}$$

Prof. This is easily proved using the Lorentz inner product formula in Eq.(6). We can obtain the Lorentz inner product:

$$\left\langle g_{\boldsymbol{r}}^L(\begin{bmatrix} \boldsymbol{h}^\top \\ \boldsymbol{W} \end{bmatrix})\boldsymbol{r}, g_{\boldsymbol{r}}^L(\begin{bmatrix} \boldsymbol{h}^\top \\ \boldsymbol{W} \end{bmatrix})\boldsymbol{r} \right\rangle_{\mathcal{L}} = 1/K \tag{17}$$

so we can prove that: $\mathbb{V}_h^{\mathrm{L}} \in \mathbb{L}_K^d$.

MRME-KGC maps the embeddings of $\mathbb{E}_K$, $\mathbb{P}_K$, $\mathbb{L}_K$ and $\mathbb{S}_K$ to the tangent space through logarithmic mapping and employs an attention mechanism to better capture the geometric information of each view. Finally, the predicted tail entity t_p is obtained through the prediction function as follows:

$$\begin{aligned} t_p &= \phi\left(\mathbb{V}_h^E, \mathbb{V}_h^P, \mathbb{V}_h^L, \mathbb{V}_h^S\right) \\ &= \odot\left(\alpha_E^h \mathbb{V}_h^E, \alpha_P^h \log_x^k\left(\mathbb{V}_h^P\right), \alpha_L^h \log_x^k\left(\mathbb{V}_h^L\right), \alpha_S^h \log_x^k\left(\mathbb{V}_h^S\right)\right) \end{aligned} \tag{18}$$

where $\odot$ represents the concat operation, it connects the embeddings in different view spaces along the specified dimension. $\boldsymbol{\alpha}$ is the attention vector and $(\alpha_E^h, \alpha_P^h, \alpha_L^h, \alpha_S^h) = \mathrm{Softmax}\left(\alpha^T\mathbb{V}_h^E, \alpha^T\mathbb{V}_h^P, \alpha^T\mathbb{V}_h^L, \alpha^T\mathbb{V}_h^S\right)$. Typically, when not considering the extension of Euclidean space to Riemannian manifolds, the process of propagating information from the head entity to the tail entity and vice versa is the same (e.g., in models like TransE [16]). However, previous work [27] has pointed out that fundamental operations in hyperbolic space, such as Möbius addition, do not satisfy commutativity. This indicates that the information propagation from the head entity to the tail entity and from the tail entity to the head entity differs in direction. To address this issue, we enhance the information transmission in Riemannian spaces by simultaneously utilizing the embeddings of head and tail entities from different Riemannian spaces when constructing the score function. Get the head entity h_p given the tail entity and relation prediction in the same way as Eq.(19):

$$\begin{aligned} h_p &= \phi\left(\mathbb{V}_t^E, \mathbb{V}_t^P, \mathbb{V}_t^L, \mathbb{V}_t^S\right) \\ &= \odot\left(\alpha_E^t \mathbb{V}_t^E, \alpha_P^t \log_x^k\left(\mathbb{V}_t^P\right), \alpha_L^t \log_x^k\left(\mathbb{V}_t^L\right), \alpha_S^t \log_x^k\left(\mathbb{V}_t^S\right)\right) \end{aligned} \tag{19}$$

the meaning of the formula characters in Eq.(19) is similar to that in Eq.(18). Where, $\mathbb{V}_t^E, \mathbb{V}_t^P, \mathbb{V}_t^L, \mathbb{V}_t^S$ are obtained by functions $g_r^{\mathbb{E}}(\mathbf{t}), g_r^{\mathbb{P}}(\mathbf{t}), g_r^{\mathbb{L}}(\mathbf{t}), g_r^{\mathbb{S}}(\mathbf{t})$ respectively, r' represents the inverse relation of r.

After integrating the head and tail entity information from different spaces, we designed the scoring function of MRME-KGC, which simultaneously considers the two-way message passing between the head and tail entities:

$$f(h,r,t) = -\left|d_c(t_p,t) + d_c(h_p,h)\right| + \gamma \tag{20}$$

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3538110

where $d_c(\cdot)$ represents the distance function and γ is the margin deviation. Then the cross entropy loss function is used to train the predictive ability of the model:

$$\mathcal{L}_p = \sum_{(h,r,t)\in\{\mathcal{G}\cup\mathcal{G}\}} \log\left(1+\exp\left(\Theta_{(h,r,t)}\cdot f(h,r,t)\right)\right)$$
$$\text{in which, } \Theta_{(h,r,t)} = \begin{cases} 1 \text{ for } (h,r,t)\in\mathcal{G} \\ -1 \text{ for } (h,r,t)\in\mathcal{G}' \end{cases} \tag{21}$$

where $\mathcal{G}\cup\mathcal{G}'$ represents the set of positive and negative samples of the triple after a random replacement operation similar to [16]. $f(\cdot)$ represents the score function of the model.

3) Riemannian Manifolds Contrastive Learning Method: As an effective unsupervised learning method, contrastive learning learns useful feature representations by comparing the similarities and differences between samples and has achieved remarkable success in many tasks. This article explores and proposes a contrastive learning method, and hopes that it can contribute to the KGC model based on Riemannian Manifolds.

In addition to addressing the issues described in the introduction for Riemannian Manifolds models, there is another noteworthy problem. The difference between the Poincaré Ball view and the Spherical space view lies merely in their curvature; both views achieve space transformation through the same mapping operations, implying that these two views are essentially isomorphic. To mitigate the potential noise that might arise from the fusion of the Poincaré Ball view and the Spherical space view, this paper intends to use contrastive learning to increase the distance between these two view spaces, thereby reducing potential noise. To this end, this paper proposes a contrastive learning method called M-CL. The core idea is to find an anchor point in both the Poincaré Ball and the spherical space that best represents their respective characteristics, as illustrated in the model diagram (Fig. 3). This anchor point is treated as the positive sample within its respective view, while points from the other view are treated as negative samples. The purpose of this approach is to achieve better representation learning by pulling closer the distances between entities within the same space and pushing apart the distances between points from different views.

Specifically, we employ two methods to find anchor points. We define the entity sets in the Poincaré Ball view and the Spherical space view as $\mathcal{P} = \left\{\boldsymbol{e}_1^p, \boldsymbol{e}_2^p, \ldots, \boldsymbol{e}_{|\mathcal{P}|}^p\right\}$ and $\mathcal{S} = \left\{\boldsymbol{e}_1^s, \boldsymbol{e}_2^s, \ldots, \boldsymbol{e}_{|\mathcal{S}|}^s\right\}$ respectively. In the first method, M_1, the weighted averages of the respective entity sets are calculated and considered as the anchor points C^p and C^s. In the second method, M_2 (optimal), we utilize the K-means clustering approach, applying the distance formula in Eq.(5) to perform the clustering. The centroids of the clusters generated from the Poincaré Ball view and the Spherical space view are taken as the anchor points, yielding C^p and C^s similarly. We implement the contrastive learning method using the InfoNCE loss function:

$$\mathcal{L}_c = \sum_{e\in\mathcal{E}} -\log\frac{\exp\left(\mathbf{e}\cdot\mathbf{c}_i/\tau\right)}{\sum_{\mathbf{c}\in\mathrm{C}\neq i}\exp\left(\mathbf{e}\cdot\mathbf{c}_j/\tau\right)+\exp\left(\mathbf{e}\cdot\mathbf{c}_i/\tau\right)} \tag{22}$$

where $\mathrm{C}\in\mathrm{C}^p\cup\mathrm{C}^s$, $\mathcal{E}=\mathcal{S}\cup\mathcal{P}$ and τ represents the temperature parameter in contrastive learning.

4) Loss Function and Training: To train the MRME-KGC model proposed in this paper, the total loss function $\mathcal{L}$ of the model consists of two parts:

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_c \tag{23}$$

Among them, $\mathcal{L}_p$ and $\mathcal{L}_c$ represent the cross entropy loss function and the contrastive learning loss function used for prediction, respectively. Furthermore, consistent with most Riemannian manifold-based works [25] [27] [22], MRME-KGC employs the N3 regularization method and Adagrad [55] as the optimizer. Additionally, the model follows the approach in [16], where negative samples are generated by randomly corrupting either the head or tail entities and then filtering the results to form the training set.

IV. EXPERIMENT AND ANALYSIS

In this section, we first elaborate on the setup and procedure of the experiment in detail. Next, we conduct a comparative analysis of the MRME-KGC model proposed in this paper against state-of-the-art KGC baseline models on multiple datasets for the main link prediction experiment, to validate the effectiveness of the MRME model. Additionally, we perform extensive supplementary experiments on the MRME-KGC model to evaluate various performance metrics. These experiments include ablation studies, fine-grained relationship experiments, experiments under low-dimensional dense features, studies on the impact of embedding dimensions on model performance, model training convergence experiments, and parameter sensitivity analysis, to understand the characteristics and performance of the MRME-KGC model comprehensively.

A. Experiment Setup

1) Datasets: We conducted extensive link prediction experiments on six different types of benchmark datasets, including three classic datasets: FB15K-237, WN18RR, YAGO3-10, a commonsense dataset CN100K, and two small datasets from the domains of social relationships and biomedicine: Kinships and UMLS. Detailed statistics of these datasets can be found in Table II.

- FB15K-237 [56], WN18RR [40], and YAGO3-10 [57] are standard datasets widely used for knowledge graph completion. YAGO3-10 is a subset extracted from YAGO3, focusing on entities involved in at least ten different relations, making it suitable for evaluating the ability to handle complex interactions and multiple relations. FB15K-237 and WN18RR are improved versions of the original datasets [16], designed to reduce test set leakage by removing redundant and reversible relations.
- The CN-100K [60] dataset originates from ConceptNet, a large multilingual commonsense knowledge graph.

TABLE II
DATASET STATISTICS.

Dataset	#Entity	#Relation	#Train	#Valid	#Test	ξ_G
WN18RR	40,943	11	86,835	3,3034	3,314	-2.54
FB15K-237	14,541	237	272,115	17,535	20,466	-0.65
YAGO3-10	123,182	37	1,079,040	5,000	5,000	-0.54
Kinships	104	26	8,544	1,068	1,074	-
UML-S	135	46	5,216	652	661	-
CN-100K	78,334	34	100,000	1,200	1,200	-

ConceptNet contains a diverse range of commonsense knowledge. The number of nodes in CN-100K is several orders of magnitude larger than in conventional datasets such as FB15K-237. Moreover, CN-100K is much sparser than traditional KGs (FB15K-237), which increases the difficulty of training.

- Kinships [58] and UMLS [59] are two small datasets. Kinships contain social network information from indigenous Australian communities, reflecting complex family structures and social relationships, while UMLS covers biomedical terminology, encompassing multiple domains such as drugs, diseases, treatments, and anatomical structures.

2) Baselines: We compare our experimental results with the following state-of-the-art models: TransE [16], ExpressivE [19], VLP [61], CompoundE [20], GreenKGC [18], Mixup [21] , HAKE-SymCL [63], TDN [64], HyboNet [47], M2GCN [25], GIE [27], RESCAL-FFHR [65], NFE [24], TransERR [66], FFTRotH [67], UniGE [68], and MGTCA [69], among the main experimental baseline models. Additionally, we include baseline models from other experiments such as RelEns-DSC [62], DRGI [71], LorentzKG [75] and OERL [70].

3) Evaluation metrics: To evaluate the performance of MRME-KGC on the KGC task, we employ four key metrics: Mean Reciprocal Rank (MRR) and Hits@1, Hits@3, and Hits@10. MRR is a metric used to assess the ranking quality of the prediction results. For each test query (typically a triplet in a missing link prediction task), the model generates a ranked list of candidate entities. MRR is the average of the reciprocal ranks of these prediction results. Hits@N is another widely used evaluation metric in the KGC domain. It measures the proportion of correct entities ranked within the top N positions by the KGC model. This metric is particularly useful for assessing the ranking performance of models in predicting missing links in knowledge graphs. The definitions of MRR and Hits@N are as follows:

$$\mathbf{MRR} = \frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{\text{rank}i} = \frac{1}{|S|}\left(\frac{1}{\text{rank1}}+\frac{1}{\text{rank2}}+\ldots+\frac{1}{\text{rank}|S|}\right) \tag{24}$$

where S denotes the set of triples, $|S|$ represents the number of triples in the set, and $rank_i$ is the link prediction rank of the i-th triple.

$$\mathbf{Hits}@N = \frac{1}{|S|}\sum_{i=1}^{|S|}\mathbb{I}\left(\text{rank}_i \leq n\right) \tag{25}$$

the function II$(\cdot)$ employed in our analysis is an indicator function, which outputs a value of 1 if the specified condition is satisfied and 0 otherwise. In our experiments, the indicator function is evaluated at different thresholds, commonly using values of n such as 1, 3, or 10. These values correspond to common metrics in KGC tasks, specifically Hits@1, Hits@3, and Hits@10, respectively.

4) Implementation Detail: We implemented the MRME-KGC model using the PyTorch library [72], the Riemannian methods library geoopt [53][1], and Adagrad [55] as the model optimizer. All computational experiments in this paper were conducted on a Linux server equipped with an Intel Xeon Gold 6226R processor (2.90GHz) and four Nvidia GTX 3090 GPUs. The model hyperparameters were selected through grid search to determine the optimal choices for MRME-KGC on each dataset, based on a combined evaluation of MRR and Hits@N. Due to the GPU memory limitations of the server, the embedding dimension for MRME-KGC was restricted to 50 dimensions on the YAGO3-10 dataset, while 100 dimensions were chosen for the other datasets. For future research, the MRME-KGC source code of the model is available on GitHub[2].

B. Main Results

To validate the intuitive effectiveness of the MRME-KGC model, we report the comparative performance of MRME-KGC against relevant baseline models in this section. The experimental results of MRME-KGC on six different types of datasets are presented in Tables III, IV, and V. Table VI shows the experimental results of MRME-KGC under low-dimensional conditions (d=32).

1) Experimental results of the MRME-KGC model on mainstream KG datasets: From Table III, it is evident that our proposed MRME-KGC model achieves superior performance compared to the most recent state-of-the-art baseline models. Specifically, in comparison to the GIE model, which also considers multiple spaces, MRME-KGC achieves improvements of 24.31%, 39.48%, 18.70%, and 6.88% on the FB15K-237 dataset in terms of MRR, Hits@1, Hits@3, and Hits@10, respectively. On the WN18RR dataset, MRME-KGC shows enhancements of 12.02%, 12.17%, 12.28%, and 10.96% across the same metrics. Similarly, on the YAGO3-10 dataset, the improvements are 17.16%, 23.43%, 13.73%, and 8.25%, respectively. These results indicate that simultaneously considering the Lorentz model and Poincaré ball model views can effectively capture the hierarchical structure of KGs, leading to better performance.

Additionally, as shown in Table III, MRME-KGC improves by 1%-29% across various metrics compared to the current state-of-the-art baseline models. Notably, MRME-KGC achieves optimal results with only 100 dimensions for the FB15K-237 and WN18RR datasets and 50 dimensions for the YAGO3-10 dataset, whereas other baseline models (UniGE, MGTCA, etc.) typically use 200-500 dimensions. These findings demonstrate that MRME-KGC can encode more semantic information with fewer dimensions, highlighting the necessity of using multiple spatial views to represent KGs. Moreover,

[1]https://github.com/geoopt/geoopt
[2]https://github.com/2391134843/MRME-KGC

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3538110

TABLE III
LINK PREDICTION MAIN RESULTS OF MRR AND HITS@K ON FB15K-237, YAGO3-10, AND WN18RR DATASETS. THE BEST SCORE IS IN **BOLD** AND THE SECOND BEST SCORE IS <u>UNDERLINED</u>. GAINS REPRESENT THE IMPROVEMENT RATIO COMPARED TO THE SUBOPTIMAL RESULT. THE EXPERIMENTAL RESULTS OF THE BASELINE MODELS ARE TAKEN FROM THEIR RESPECTIVE ORIGINAL PAPERS.

Model	FB15K-237				WN18RR				YAGO3-10			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
Euclidean approaches												
ExpressivE(2023)	0.333	24.3	36.6	51.2	0.464	46.4	52.2	59.7	-	-	-	-
VLP(2023)	0.362	27.1	39.7	54.2	0.498	45.5	51.4	58.2	-	-	-	-
CompoundE(2023)	0.367	27.5	40.2	55.5	0.493	0.451	0.507	0.578	0.491	45	50.8	57.6
GreenKGC(2023)	0.345	26.5	36.9	50.7	0.345	36.7	43	49.1	0.453	36.1	50.9	62.9
RelEns-DSC(2023)	0.368	27.4	<u>40.4</u>	55.5	<u>0.520</u>	<u>47.7</u>	<u>53.7</u>	<u>60.3</u>	-	-	-	-
HAKE-SymCL(2024)	0.346	24.8	38,4	54.4	0.497	45.4	51.5	58.5	-	-	-	-
TDN(2023)	0.358	27.3	40.3	56,1	0.499	45.5	52.3	57.9	-	-	-	-
Mixup(2023)	0.281	-	-	45.3	0.422	-	-	48.8	-	-	-	-
Non-Euclidean approaches												
HyboNet(2022)	0.352	26.3	38.7	52.9	0.513	48.2	52.7	56.9	-	-	-	-
M2GNN(2021)	0.362	27.5	39.8	56.5	0.485	44.4	49.8	57.2	0.543	47.8	60.5	70.2
RESCAL-FFHR(2023)	0.345	25.6	37.9	52.1	0.468	42.2	49.0	55.2	-	-	-	-
TransERR(2023)	0.36	26.4	39.6	55.5	-	-	-	-	0.49	40.4	53.8	64.7
FFTRotH(2022)	0.319	22.8	35.2	50	0.484	43.7	50.2	57.2	-	-	-	-
LorentzKG(2024)	0.384	28.7	42.2	57.9	0.502	45.6	52.3	58.9	-	-	-	-
GIE(2022)	0.362	27.1	40.1	55.2	0.491	45.2	50.5	57.5	0.579	50.5	61.8	70.9
NFE(2023)	0.355	26.1	-	54.3	0.483	43.8	-	57.6	0.570	-	49.8	69.7
UniGE(2024)	0.357	26.4	39.1	55.9	0.502	45.5	52	59.2	58.3	51.2	62.7	71.5
MGTCA(2024)	<u>0.393</u>	<u>29.1</u>	40.1	<u>58.3</u>	0.511	47.5	52.5	59.3	<u>0.586</u>	<u>51.4</u>	<u>62.9</u>	<u>72.1</u>
Our methods												
MRME-KGC(ours)	**0.45**	**37.8**	**47.6**	**59**	**0.55**	**50.7**	**56.7**	**63.8**	**68.3**	**63.2**	**71.3**	**77.4**
Gains(%)	*14.50*	*29.80*	*17.80*	*1.20*	*5.70*	*5.10*	*5.50*	*5.45*	*17*	*22*	*13.30*	*7.30*

we observe a significant improvement in the Hits@1 metric, which is due to most models' inability to handle the complex structure of KGs effectively, resulting in less accurate predictions. MRME-KGC, by capturing structural information across multiple views, can learn more reliable KG representations, leading to more accurate Hits@1 results.

2) Experimental results of the MRME-KGC model on Common-Sense KG datasets : To further validate the effectiveness of MRME-KGC from multiple perspectives, we selected the commonsense KG dataset CN-100K for model evaluation. Unlike mainstream KG datasets (such as FB15K-237), the commonsense KG dataset has more nodes, making the entire KG sparser (with an average in-degree of 16.98 for FB15K-237 compared to only 1.25 for CN-100K) [73]. Generally, sparser KGs are more challenging to learn from, so we chose the CN-100K dataset for our experiments.

Table IV reports the experimental results of MRME-KGC on the CN-100K dataset. Compared to GIE, MRME-KGC improves by 24.64%, 38.94%, 6.90%, and 5.29% in terms of MRR, Hits@1, Hits@10, and Hits@100, respectively. Additionally, it is evident that earlier KGC methods, such as TransE and RotatE, struggle significantly with the CN-100K dataset, resulting in generally low-performance metrics. Even models like OERL [70] fall significantly behind MRME-KGC in the Hits@1 metric. These observations strongly demonstrate the superiority of MRME-KGC, as it consistently achieves state-of-the-art results even when faced with more complex KG datasets.

3) Experimental results of the MRME-KGC model on Kinships and UMLS KG datasets: To evaluate MRME-KGC's performance across a broader range of KG datasets, we selected two classic small datasets: Kinship and UMLS. Table V summarizes the comparison between our MRME-KGC model and baseline models on these simpler, less complex datasets. On Kinship, MRME-KGC achieved the best results across all metrics, particularly excelling in Hits@1 with a 22.22% improvement over TDN. It also showed increases of 12.68% in

TABLE IV
EXPERIMENTAL RESULTS ON THE CN-100K DATASET. ♣ REPRESENTS THE RESULT OF THE REPRODUCTION. THE RESULTS OF OTHER BASELINE MODELS ARE TAKEN FROM [70]

Model	MRR			H@1			H@10			H@100		
	Head	Tail	Avg	Head	Tail	Avg	Head	Tail	Avg	Head	Tail	Avg
TransE	16.2	0.171	16.69	2.33	2.00	2.17	41.67	43.92	42.79	67.83	67.50	67.67
TransD	14.97	15.02	14.99	1.67	1.67	1.67	37.83	39.67	38.75	63.17	64.00	63.58
ComlEx	14.83	13.16	13.99	6.83	5.58	6.21	31.67	28.42	30.04	60.25	57.42	58.83
RotatE	20.06	21.99	21.02	6.92	8.50	7.71	44.33	48.42	46.38	71.67	73.17	72.42
ConvE	21.04	23.78	22.41	13.00	14.33	13.67	37.50	42.42	39.96	66.17	66.00	66.08
OERL+MLP	31.25	33.16	32.20	20.75	22.33	21.54	54.08	55.58	54.83	83.00	75.50	79.25
OERL+Mean	34.75	39.25	37.00	22.33	28.50	25.42	60.50	60.67	60.58	86.75	77.50	82.13
OERL	36.57	40.33	38.45	24.42	29.08	26.75	63.00	62.25	62.63	**88.50**	77.67	83.08
GIE♣	-	-	56.00	-	-	47.6	-	-	71.60	-	-	83.42
MRME-KGC	**66.50**	**73.10**	**69.80**	**63.50**	**68.74**	**66.12**	**74.00**	**79.08**	**76.54**	85.17	**90.49**	**87.83**

TABLE V
EXPERIMENTAL RESULTS ON THE KINSHIP AND UML DATASET. THE RESULTS OF THE BASELINE MODELS ARE TAKEN FROM [64]. THE BEST RESULT AND THE SECOND-BEST RESULT ARE INDICATED IN BOLD AND UNDERLINED RESPECTIVELY.

Model	Kinship				UMLS			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	0.211	9.3	25.2	47.00	0.615	39.1	80.7	94.5
RotatE	0.738	61.7	82.7	95.4	0.4433	48.42	46.38	71.67
ConvE	0.772	66.5	85.8	95.0	0.836	76.4	91.7	94.6
DRGI	0.847	0.981	91.5	76.5	0.898	83.8	94.8	98.8
TND(TransE)	0.858	77.5	93.4	98.5	0.961	92.6	98.3	99.9
TND(DistMult)	<u>0.867</u>	<u>78.8</u>	<u>94.2</u>	<u>98.6</u>	<u>0.963</u>	<u>92.6</u>	<u>98.5</u>	**99.9**
MRME-KGC	**0.977**	**96.3**	**98.8**	**99.4**	**0.976**	**96.1**	**99.0**	<u>99.7</u>

MRR, 4.88% in Hits@3, and 0.81% in Hits@10. For UMLS, MRME-KGC performed best in MRR, Hits@1, and Hits@3, with improvements of 1.35%, 3.78%, and 0.51%, respectively, and was only 0.2% behind TND in Hits@10. These results demonstrate MRME-KGC's superiority, especially in Hits@1, emphasizing the importance of considering multiple views simultaneously.

4) Experimental results under low-dimensional conditions on mainstream datasets: To evaluate the effectiveness of MRME-KGC in a low-dimensional (d=32) setting, we followed the experimental setup in fellow [68] and conducted experiments on three datasets. Table VI presents a comprehensive comparison between MRME-KGC and Riemannian-based baselines of the same type. It is evident that MRME-KGC

TABLE VI
LINK PREDICTION RESULTS (%) ON WN18RR, FB15K-237 AND YAGO3-10 FOR LOW-DIMENSIONAL EMBEDDINGS ($d = 32$). ALL BASELINE RESULTS ARE TAKEN FROM [68]. THE BEST SCORES ARE IN BOLD.

Model	WN18RR				FB15k-237				YAGO3-10			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
TransE	36.6	27.4	43.3	51.5	29.5	21.0	32.2	46.6	-	-	-	-
RotatE	38.7	33.0	41.7	49.1	29.0	20.8	31.6	45.8	-	-	-	-
ComplEx	42.1	39.1	43.4	47.6	28.7	20.3	31.6	45.6	33.6	25.9	36.7	48.4
MuRE	45.8	42.1	47.1	52.5	31.3	22.6	34.0	48.9	23.8	18.7	31.7	47.8
MuRP	46.5	42.0	48.4	54.4	32.3	23.5	35.3	50.1	23.0	15.0	24.7	39.2
RotH	47.2	42.8	49.0	55.3	31.4	22.3	34.6	49.7	33.4	26.4	41.5	55.9
RefH	44.7	40.0	46.4	51.8	32.2	23.4	36.0	50.4	36.3	30.6	43.5	56.3
AttH	46.1	40.9	49.0	55.1	33.0	23.9	36.0	51.4	39.7	31.0	44.7	57.6
UltraE	48.8	44.0	50.3	55.8	33.6	24.7	37.1	51.4	38.0	31.8	44.4	57.2
GIE	43.4	41.0	44.2	48.2	33.0	24.3	36.2	50.5	43.1	34.7	45.9	57.7
UniGE	49.1	44.7	51.2	**56.3**	34.5	25.7	37.5	52.3	41.2	35.2	**51.4**	57.9
MRME-KGC	**49.8**	**46.8**	**51.9**	56.2	**35.9**	**28.6**	**38.3**	**52.4**	**46.5**	**37.6**	49.7	**59.7**

performs excellently across all datasets and most evaluation metrics, particularly showing a 4%-11% improvement in MRR and Hits@1 compared to UniGE. In the YAGO3-10 dataset, although the Hits@3 metric is slightly inferior to UniGE, MRME-KGC outperforms all other metrics. This indicates that MRME-KGC maintains high prediction accuracy in low-dimensional scenarios, surpassing existing baseline models.

C. Experiments on Model Training Processes

To further understand the changes in metrics during the model training process, we conducted extensive experiments and plotted Fig. 4 to demonstrate the superiority of MRME-KGC. Fig. 4 illustrates the changes in Hits@1, Hits@3, Hits@10, and MRR for MRME-KGC compared to baseline models based on Riemannian Manifolds (AttH, RotH, GIE) across four datasets: FB15k-237, WN18RR, CN100K, and UMLS. Overall, MRME-KGC consistently outperforms the baseline models in MRR and Hits@N across all datasets, demonstrating advantages such as faster convergence, higher performance more quickly, and greater stability with fewer fluctuations during training.

In addition, several interesting phenomena warrant discussion. First, like the baseline models, MRME-KGC shows rapid improvements in metrics within the first 20-30 epochs across all datasets, indicating that the architecture and learning mechanisms of Riemannian Manifolds-based KGC models are effective in capturing relational information early in training. Second, all models exhibit steeper slopes in their training curves for the UMLS dataset, suggesting faster semantic information acquisition from simpler datasets, leading to faster convergence. Third, although the learning speed on the FB15k-237 dataset is slower, MRME-KGC still surpasses the baseline models in the end; this may be due to a lower learning rate used during the FB15k-237 experiments compared to the baseline models. Finally, the excellent results of MRME-KGC show that it is very important to consider multiple views at the same time for the KGC task, which can not only speed up the model convergence efficiency, but also achieve better performance.

D. Experiments and Analysis of Model Dimensionality and Parameter Sensitivity

To evaluate the robustness and stability of the model, we conducted a series of comparative experiments with baseline models and parameter sensitivity experiments. The main parameters we studied include the embedding dimension and the comparative learning parameter τ.

1) Comparative experiments with baseline models in different dimensions: We evaluated the performance of the MRME-KGC model and two baseline models (GIE and RTTH) on two datasets (WN18RR and FB15K-237), focusing on four metrics: Hits@10, Hits@3, Hits@1, and MRR. The baseline models AttH [22], RotH [22], and GIE [27] were reproduced using their respective open-source GitHub addresses. Some other models did not publish their code addresses and could not be used for experiments. The analysis covered various embedding dimensions (16, 32, 50, 64, 100). As shown in Fig. 5, MRME-KGC consistently outperforms the baseline models across different dimensions. Notably, its performance is particularly strong at higher dimensions (64, 100), demonstrating its robust capability in capturing complex relationships within KGs. These results also confirm that MRME-KGC maintains high accuracy and predictive power even with varying embedding dimensions.

2) Comparative learning parameter τ sensitivity experiment : As shown in Fig. 6, we analyzed the impact of the contrastive learning parameter τ on the MRME-KGC model across four metrics on two datasets. The results indicate that τ has minimal effect on MRR, Hits@1, and Hits@3. However, Hits@10 is somewhat affected by variations in τ, likely due to Hits@10 being a less precise metric that measures the hit rate within the top 10 predictions. When the model fails to accurately predict the top 1 correct entity, fluctuations in τ influence the distances in the Poincaré disk and sphere space. Overall, the experimental results demonstrate that the MRME-KGC model exhibits low sensitivity to the τ parameter, with performance remaining relatively stable across different τ values, indicating strong robustness.

E. Ablation Experiment

Table VII reports detailed ablation experiments on the MRME-KGC model, where M_1 represents the experimental results using the M_1 method for anchor point computation

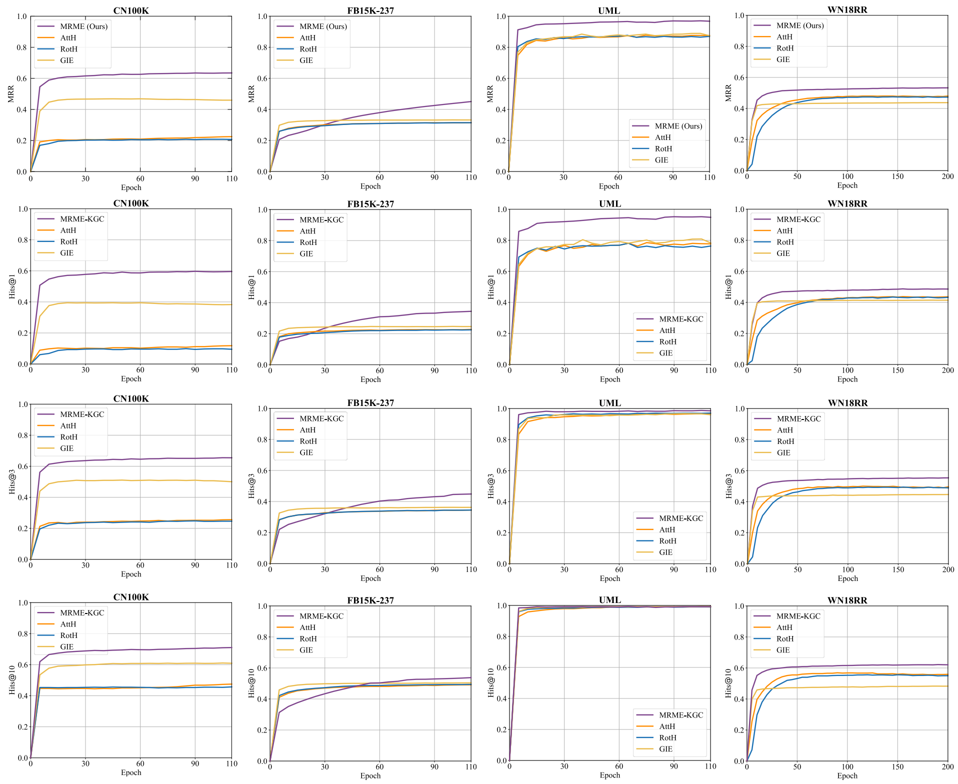


Fig. 4. Training processes compared with MRR and Hits@1,3,10 on FB15k-237 , WN18RR ,CN100K and UML datasets.

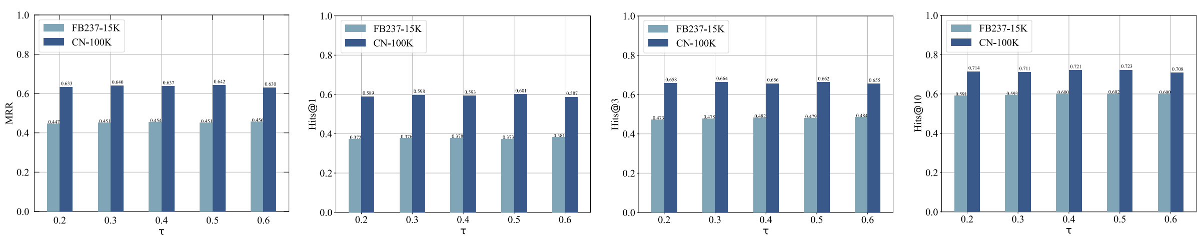


Fig. 5. Parameter sensitivity experimental results of the comparative learning parameter τ on the FB15K-237 and CN100K datasets.

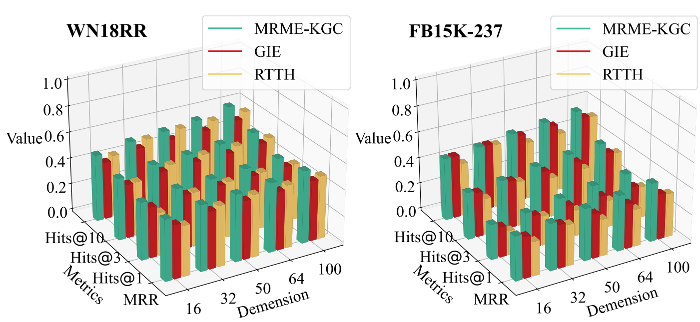


Fig. 6. Comparison of experimental results with baseline models under different dimensionality changes on FB15K-237 and WN18RR datasets.

in contrastive learning. Overall, our proposed modules are each effective, as shown by the percentage decrease relative to MRME-KGC upon ablating each component.

First, we observe that the model performance significantly drops when the Lorentz model is removed, indicating the necessity of considering both the Poincaré ball and the Lorentz model in hyperbolic modeling as we proposed. Secondly, removing the contrastive learning module results in a performance decline, demonstrating that our proposed M-CL better models complex KG and alleviates boundary issues in the Poincaré ball view and sphere view. Thirdly, we conducted ablation experiments to assess the impact of removing the Euclidean, Poincaré Ball Model, and Sphere spaces on the performance of MRME-KGC. The experimental results indicate that the Mean Reciprocal Rank (MRR) metric of the model decreased by approximately 3% after the removal of these spaces, and other metrics also showed a decline. These findings confirm that each space uniquely contributes to enhancing the model's representation and performance in KGC tasks. Finally, we note that the performance significantly decreases when the attention mechanism is removed. This suggests that simply concatenating embeddings from the four views increases noise, preventing the model from learning the semantic information inherent in the complex structure of KGs. It also highlights that the attention mechanism effectively integrates representations from multiple views.

TABLE VII
IN THE ABLATION EXPERIMENTS ON THE FB15K-237 AND WN18RR DATA SETS, WE GRADUALLY DESTROYED THE LORENTZ MODEL AND THE CONTRASTIVE LEARNING MODULE. (%) REPRESENTS THE PERCENTAGE OF EFFECT REDUCTION.

	FB15K-237				**WN18RR**			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
MRME-KGC-M_2	0.45	37.8	47.6	59.0	0.55	50.7	56.7	63.8
W/0 Lorentz	0.368(18.22%)	28.2(25.40%)	40.6(14.71%)	56.3(4.58%)	0.494(9.46%)	45.9(10.18%)	51.4(9.35%)	59.0(7.53%)
W/0 Contrastive learning	0.443(1.56%)	36.3(3.97%)	46.1(3.15%)	57.8(2.03%)	0.531(3.45%)	49.8(1.77%)	54.9(3.17%)	62.2(2.51%)
MRME-KGC-M_1	0.441(2.00%)	36.0(4.76%)	46.3(2.73%)	57.5(2.54%)	0.529(3.82%)	49.1(3.15%)	54.6(3.70%)	61.7(3.29%)
W/O attention fusion	0.378(16%)	26.6(29.63%)	39.1(17.86%)	42.7(27.63%)	0.465(15.45%)	42.57(16.03%)	47.97(15.36%)	54.26(14.96%)
W/0 Euclidean	0.435(3.31%)	36.24(4.12%)	45.83(3.74%)	57.58(2.40%)	0.534(2.94%)	48.99(3.37%)	55.00(3.05%)	62.17(2.53%)
W/0 Poincaré Ball Model	0.436(3.05%)	36.36(3.78%)	46.04(3.27%)	57.89(1.88%)	0.539(1.96%)	49.75(1.89%)	55.39(2.29%)	62.53(2.02%)
W/0 Sphere	0.435(3.31%)	36.12(4.47%)	46.00(3.50%)	57.68(2.23%)	0.536(2.56%)	49.47(2.43%)	55.20(2.64%)	62.47(2.08%)

TABLE VIII
FINE-GRAINED EXPERIMENTAL RESULTS OF EACH RELATION ON THE HITS@10 METRIC ON THE WN18RR DATASET. THE BEST RESULT AND THE SECOND-BEST RESULT ARE INDICATED IN BOLD AND UNDERLINED RESPECTIVELY.

Relation Name	Khs_G	ξ_G	RotatE	QuatE	CompoundE	MuRMP	GIE	UniGE	MRME-KGC
also_see	0.36	-2.09	0.627	0.607	0.629	0.72	0.759	<u>0.768</u>	**0.787**
derivationally_related_form	0.4	-3.84	0.957	0.952	0.956	<u>0.97</u>	0.968	0.964	**0.9767**
has_part	1	-1.43	0.205	0.21	0.2	0.316	0.334	<u>0.341</u>	**0.3779**
hypernym	1	-2.46	0.154	0.172	0.179	0.232	0.262	<u>0.274</u>	**0.3813**
instance_hypernym	1	-0.82	0.324	0.362	0.351	0.491	0.501	<u>0.511</u>	**0.8443**
member_meronym	1	-2.9	0.255	0.236	0.254	0.35	<u>0.36</u>	0.357	**0.4506**
member_of_domain_region	1	-0.78	0.243	0.14	0.401	0.349	<u>0.404</u>	**0.437**	0.431
member_of_domain_usage	1	-0.74	0.277	0.372	0.309	0.42	<u>0.438</u>	0.437	**0.442**
similar_to	0.07	-1	1	1	1	1	1	1	**1**
synset_domain_topic_of	0.99	-0.69	0.334	0.395	0.382	0.445	0.435	<u>0.446</u>	**0.7807**
verb_group	0.07	-0.5	0.968	0.93	0.974	0.981	**0.984**	0.981	**0.984**

F. Fine-Grained Experiments on Each Relationship

To investigate the model's performance on individual relations, we conducted fine-grained experiments on WN18RR. Two primary metrics assess the hierarchical structure of relations: curvature estimation ξ_G (indicating the tree-like hierarchy of the graph) and Krackhardt hierarchy score Khs_G (indicating the number of small cycles in the graph). Curvature estimation captures global hierarchical behavior, while the Krackhardt score captures local behavior. Detailed calculations for these metrics are provided in the [22] [23].

From Table VIII, several observations can be made: First, in highly hierarchical settings, Riemannian Manifolds-based models outperform those based on Euclidean/complex space. Second, MRME-KGC achieved the best performance in 10 out of 11 relations. Additionally, results on the $instance\_hypernym$ relation indicate that the model performs well even in less hierarchical structures. Lastly, MRME-KGC improved performance by approximately 22.31% over GIE and 20.28% over UniGE [68] across the 11 relations. These findings demonstrate that MRME-KGC significantly enhances performance over baseline models in handling various fine-grained relations. Overall, MRME-KGC's fusion of four views combined with contrastive learning yields impressive results, performing well across different relations.

V. CONCLUSION

In this paper, we propose a knowledge graph completion model based on Riemannian Manifolds fusion and contrastive learning, named MRME-KGC. MRME-KGC enhances knowledge graph modeling by simultaneously considering four perspectives: two negative-curvature hyperbolic Riemannian spaces, a zero-curvature Euclidean Riemannian space, and a positive-curvature spherical Riemannian space. The fusion of these four perspectives strengthens the model's capability. Additionally, we introduce a contrastive learning method tailored for Riemannian spaces to mitigate noise and representation issues arising from multi-view Riemannian manifold fusion. Furthermore, the model's training is improved through a bidirectional scoring function. Extensive experiments on six different types of datasets demonstrate that MRME-KGC achieves highly effective performance.

In future work, we aim to explore the potential of more complex operations on Riemannian Manifolds within the context of KGC. In addition, we will investigate the integration of the MRME-KGC model into dynamic knowledge graphs and multimodal knowledge graphs.

REFERENCES

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 2, pp. 494–514, 2021.

[2] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, F. Sun, and K. He, "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024.

[3] Y. Li, L. Hou, and J. Li, "Towards knowledge-aware and deep reinforced cross-domain recommendation over collaborative knowledge graph," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[4] L. Chen, J. Cao, H. Tao, and J. Wu, "Trip reinforcement recommendation with graph-based representation learning," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 4, pp. 1–20, 2023.

[5] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[6] R. Zhang, Y. Su, B. D. Trisedya, X. Zhao, M. Yang, H. Cheng, and J. Qi, "Autoalign: fully automatic and effective knowledge graph alignment enabled by large language models," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[7] Y. Ding, A. Poudel, Q. Zeng, T. Weninger, B. Veeramani, and S. Bhattacharya, "Entgpt: Linking generative large language models with knowledge bases," *arXiv preprint arXiv:2402.06738*, 2024.

[8] Z. Zhao, Y. Jing, F. Feng, J. Wu, C. Gao, and X. He, "Leave no patient behind: Enhancing medication recommendation for rare disease patients," *arXiv preprint arXiv:2403.17745*, 2024.

[9] J. Shi, Z. Yuan, W. Guo, C. Ma, J. Chen, and M. Zhang, "Knowledge-graph-enabled biomedical entity linking: a survey," *World Wide Web*, vol. 26, no. 5, pp. 2593–2622, 2023.

[10] D. Bang, S. Lim, S. Lee, and S. Kim, "Biomedical knowledge graph learning for drug repurposing by extending guilt-by-association to multiple layers," *Nature Communications*, vol. 14, no. 1, p. 3570, 2023.

[11] M. Liu, K. Liang, Y. Zhao, W. Tu, S. Zhou, X. Gan, X. Liu, and K. He, "Self-supervised temporal graph learning with temporal and structural intensity alignment," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[12] X. Li, Y. Li, X. Mo, H. Xiao, Y. Shen, and L. Chen, "Diga: Guided diffusion model for graph recovery in anti-money laundering," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4404–4413.

[13] Y. Li, Z. Zhu, L. Chen, B. Yang, Y. Wu, X. Guo, B. Han, and Y. Zhao, "Exploring large-scale financial knowledge graph for smes supply chain mining," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[14] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[15] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[16] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[17] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[18] Y. C. Wang, X. Ge, B. Wang, and C.-C. J. Kuo, "GreenKGC: A lightweight knowledge graph completion method," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 10 596–10 613. [Online]. Available: https://aclanthology.org/2023.acl-long.591

[19] A. Pavlović and E. Sallinger, "Expressive: A spatio-functional embedding for knowledge graph completion," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=xkev3_np08z

[20] X. Ge, Y. C. Wang, B. Wang, and C.-C. J. Kuo, "Compounding geometric operations for knowledge graph completion," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 6947–6965. [Online]. Available: https://aclanthology.org/2023.acl-long.384

[21] T. Xie and Y. Ge, "Enhance knowledge graph embedding by mixup," *IEEE Transactions on Knowledge & Data Engineering*, vol. 36, no. 02, pp. 569–580, 2024.

[22] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6901–6914.

[23] Y. Bai, Z. Ying, H. Ren, and J. Leskovec, "Modeling heterogeneous hierarchies with relation-specific hyperbolic cones," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 316–12 327, 2021.

[24] C. Xiao, X. He, and Y. Cao, "Knowledge graph embedding by normalizing flows," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4756–4764.

[25] S. Wang, X. Wei, C. N. Nogueira dos Santos, Z. Wang, R. Nallapati, A. Arnold, B. Xiang, P. S. Yu, and I. F. Cruz, "Mixed-curvature multi-relational graph neural network for knowledge graph completion," in *Proceedings of the Web Conference 2021*, 2021, pp. 1761–1771.

[26] W. Peng, T. Varanka, A. Mostafa, H. Shi, and G. Zhao, "Hyperbolic deep neural networks: A survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 10 023–10 044, 2021.

[27] Z. Cao, Q. Xu, Z. Yang, X. Cao, and Q. Huang, "Geometry interaction knowledge graph embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 5, 2022, pp. 5521–5529.

[28] L. Sun, Z. Huang, Z. Wang, F. Wang, H. Peng, and S. Y. Philip, "Motif-aware riemannian graph neural network with generative-contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, 2024, pp. 9044–9052.

[29] T. Yu and C. D. Sa, "Random laplacian features for learning with hyperbolic space," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. [Online]. Available: https://openreview.net/pdf?id=3pfNb4pZBNp

[30] J. Dai, Y. Wu, Z. Gao, and Y. Jia, "A hyperbolic-to-hyperbolic graph convolutional network," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 154–163.

[31] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 687–696.

[32] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[33] T. Ebisu and R. Ichise, "Generalized translation-based embedding of knowledge graph," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 941–951, 2019.

[34] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[35] L. Li, X. Zhang, Z. Jin, C. Gao, R. Zhu, Y. Liang, and Y. Ma, "Knowledge graph completion method based on quantum embedding and quaternion interaction enhancement," *Information Sciences*, vol. 648, p. 119548, 2023.

[36] Z. Zhang, J. Cai, Y. Zhang, and J. Wang, "Learning hierarchy-aware knowledge graph embeddings for link prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 03, 2020, pp. 3065–3072.

[37] W. R. Hamilton, "Lxxviii. on quaternions; or on a new system of imaginaries in algebra: To the editors of the philosophical magazine and journal," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 25, no. 169, pp. 489–495, 1844.

[38] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion knowledge graph embeddings," *Advances in neural information processing systems*, vol. 32, 2019.

[39] L. Li, X. Zhang, Y. Ma, C. Gao, J. Wang, Y. Yu, Z. Yuan, and Q. Ma, "A knowledge graph completion model based on contrastive learning and relation enhancement method," *Knowledge-Based Systems*, vol. 256, p. 109889, 2022.

[40] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[41] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," *arXiv preprint arXiv:1712.02121*, 2017.

[42] S. Vashishth, S. Sanyal, V. Nitin, N. Agrawal, and P. Talukdar, "Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 03, 2020, pp. 3009–3016.

[43] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[44] M. Yang, M. Zhou, H. Xiong, and I. King, "Hyperbolic temporal network embedding," *IEEE Transactions on Knowledge & Data Engineering*, vol. 35, no. 11, pp. 11 489–11 502, 2023.

[45] Y. Chen, M. Yang, Y. Zhang, M. Zhao, Z. Meng, J. Hao, and I. King, "Modeling scale-free graphs with hyperbolic geometry for knowledge-aware recommendation," in *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022, pp. 94–102.

[46] J. Sun, Z. Cheng, S. Zuberi, F. Pérez, and M. Volkovs, "Hgcf: Hyperbolic graph convolution networks for collaborative filtering," in *Proceedings of the Web Conference 2021*, 2021, pp. 593–601.

[47] W. Chen, X. Han, Y. Lin, H. Zhao, Z. Liu, P. Li, M. Sun, and J. Zhou, "Fully hyperbolic neural networks," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 5672–5686. [Online]. Available: https://aclanthology.org/2022.acl-long.389

[48] X. Fu, Y. Gao, Y. Wei, Q. Sun, H. Peng, J. Li, and X. Li, "Hyperbolic geometric latent diffusion model for graph generation," *arXiv preprint arXiv:2405.03188*, 2024.
[49] J. M. Lee and J. M. Lee, *Smooth manifolds*. Springer, 2012.
[50] P. Petersen, *Riemannian geometry*. Springer, 2006, vol. 171.
[51] O. Ganea, G. Bécigneul, and T. Hofmann, "Hyperbolic neural networks," *Advances in neural information processing systems*, vol. 31, 2018.
[52] A. A. Ungar, "Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry," *Computers & Mathematics with Applications*, vol. 41, no. 1-2, pp. 135–147, 2001.
[53] M. Kochurov, R. Karimov, and S. Kozlukov, "Geoopt: Riemannian optimization in pytorch," *arXiv preprint arXiv:2005.02819*, 2020.
[54] A. Bdeir, K. Schwethelm, and N. Landwehr, "Fully hyperbolic convolutional neural networks for computer vision," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=ekz1hN5QNh
[55] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.
[56] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1499–1509.
[57] F. Mahdisoltani, J. Biega, and F. M. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *CIDR*, 2013.
[58] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *AAAI*, vol. 3, 2006, p. 5.
[59] A. T. McCray, "An upper-level ontology for the biomedical domain," *Comparative and Functional genomics*, vol. 4, no. 1, pp. 80–84, 2003.
[60] X. Li, A. Taheri, L. Tu, and K. Gimpel, "Commonsense knowledge base completion," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1445–1455.
[61] R. Li, X. Chen, C. Li, Y. Shen, J. Zhao, Y. Wang, W. Han, H. Sun, W. Deng, Q. Zhang *et al.*, "To copy rather than memorize: A vertical learning paradigm for knowledge graph completion," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 6335–6347.
[62] L. Yue, Y. Zhang, Q. Yao, Y. Li, X. Wu, Z. Zhang, Z. Lin, and Y. Zheng, "Relation-aware ensemble learning for knowledge graph embedding," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 16 620–16 631.
[63] K. Liang, Y. Liu, S. Zhou, W. Tu, Y. Wen, X. Yang, X. Dong, and X. Liu, "Knowledge graph contrastive learning based on relation-symmetrical structure," *IEEE Transactions on Knowledge & Data Engineering*, vol. 36, no. 01, pp. 226–238, 2024.
[64] J. Wang, B. Wang, J. Gao, X. Li, Y. Hu, and B. Yin, "Tdn: Triplet distributor network for knowledge graph completion," *IEEE Transactions on Knowledge and Data Engineering*, 2023.
[65] W. Shi, J. Wu, X. Cao, J. Chen, W. Lei, W. Wu, and X. He, "Ffhr: Fully and flexible hyperbolic representation for knowledge graph completion," *arXiv preprint arXiv:2302.04088*, 2023.
[66] Y. Li, W. Fan, C. Liu, C. Lin, and J. Qian, "Transher: Translating knowledge graph embedding with hyper-ellipsoidal restriction," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 8517–8528.
[67] H. Xiao, X. Liu, Y. Song, G. Wong, and S. See, "Complex hyperbolic knowledge graph embeddings with fast fourier transform," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 5228–5239.
[68] Y. Liu, Z. Cao, X. Gao, J. Zhang, and R. Yan, "Bridging the space gap: Unifying geometry knowledge graph embedding with optimal transport," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 2128–2137.
[69] B. Shang, Y. Zhao, J. Liu, and D. Wang, "Mixed geometry message and trainable convolutional attention network for knowledge graph completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 8, 2024, pp. 8966–8974.
[70] Q. Li, D. Wang, S. Feng, K. Song, Y. Zhang, and G. Yu, "Oerl: Enhanced representation learning via open knowledge graphs," *IEEE Transactions on Knowledge & Data Engineering*, vol. 35, no. 09, pp. 8880–8892, 2023.
[71] S. Liang, J. Shao, D. Zhang, J. Zhang, and B. Cui, "Drgi: Deep relational graph infomax for knowledge graph completion," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 1499–1500.
[72] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
[73] C. Malaviya, C. Bhagavatula, A. Bosselut, and Y. Choi, "Commonsense knowledge base completion with structural and semantic context," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 03, 2020, pp. 2925–2933.
[74] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," *Advances in neural information processing systems*, vol. 30, 2017.
[75] X. Fan, M. Xu, H. Chen, Y. Chen, M. Das, and H. Yang, "Enhancing hyperbolic knowledge graph embeddings via lorentz transformations," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 4575–4589.
[76] M. Nickel and D. Kiela, "Learning continuous hierarchies in the lorentz model of hyperbolic geometry," in *International conference on machine learning*. PMLR, 2018, pp. 3779–3788.
[77] X. Xiong, B. Shen, H. Cai, L. Jiang, P. Hu, and Y. Wang, "Knowledge graph completion based on hyperbolic graph contrastive attention network," in *2022 IEEE International Conference on e-Business Engineering (ICEBE)*. IEEE, 2022, pp. 29–31.

Linyu Li is currently pursuing a PhD degree with the School of Computer Science, Peking University, China. His research interests include knowledge graph completion and solving various knowledge graph problems using deep learning. He also serves as a reviewer for multiple journals and conferences.

Zhi Jin (Fellow IEEE) received the B.S. degree in computing science from Zhejiang University, Hangzhou, China, and the M.S. and Ph.D. degrees in computer science and engineering from the National University of Defense Technology, Changsha, China. She is currently a Full Professor of software engineering at Peking University, Beijing, China, and the Deputy Director of the Key Laboratory of High Confidence Software Technologies of the Ministry of Education of China, Peking University. She has coauthored five books and has published more than 200 referred conference/journal papers. She is/was the Principal Investigator of more than 18 national competitive grants. Her research interests include knowledge engineering, machine learning, natural language processing, and requirements engineering. She is an Associate Editor for IEEE Transactions on Software Engineering, IEEE Transactions on Reliability, and ACM Transactions on Autonomous and Adaptive Systems , and serves on the Editorial Board of Requirements Engineering Journal and Empirical Software Engineering. She was a Standing Board Member of China Computer Federation (CCF), the Chair of CCF Technical Committee of Software Engineering during 2016–2019, and Chair of CCF Technical Committee of System Software during 2020–2023. She is a Fellow of CCF and also the Chairman of the Supervisory Board of CCF.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2025.3538110

Xuan Zhang (Member, IEEE) received the B.S. and M.S. degrees in computer science from Yunnan University, Yunnan. She received her Ph.D. degree in system analysis and integration from Yunnan University, Yunnan. She is a professor with the School of Software, Yunnan University, Yunnan, China. She is author of 4 books and more than 120 papers. She has been principal investigator for more than 30 national, provincial, and private grants and contracts. She is the core scientist of Yunnan Key Laboratory of Software Engineering. Her research interests include knowledge engineering, natural language processing, and computer vision. She is a member of Institute of Electrical and Electronic Engineers and China Computer Federation.

Haoran Duan is currently pursuing a PhD in the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His research interests include graph representation learning and self-supervised learning. He has served as a reviewer for multiple artificial intelligence conferences and journals, including KDD, ECML-PKDD, IJCNN, Neural Networks, Knowledge-Based Systems, and Neurocomputing.

Jishu Wang is currently working toward a Ph.D. degree with the School of Information Science and Engineering, Yunnan University, Kunming, China. He has authored peer-reviewed papers in IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Vehicular Technology, and IEEE Transactions on Network and Service Management. His current research interests include Deep Learning Applications, blockchain, and intelligent transportation.

Zhengwei Tao is currently a PhD candidate in the School of Computer Science, Peking University, China. His research interests include event reasoning with LLM and synthetic data construction. He has published more than 10 papers on multiple artificial intelligence conferences and journals including ACL, EMNLP et al.

Haiyan Zhao (Member, IEEE) received the B.S. and M.S. degrees from the Department of Computer Science, Peking University, in 1988 and 1991, respectively, and the Ph.D. degree from the Department of Information Engineering, University of Tokyo, in 2003. She is an Associate Professor with the School of Computer Science, Peking University. Her research interests include software engineering, requirements engineering, and programming languages and systems.

Xiaofeng Zhu (Senior Member, IEEE) received the PhD degree in computer science from the University of Queensland, Australia, and is currently with the University of Electronic Science and Technology of China. His current research interests include Big Data, artificial intelligence, and medical image analysis.