
Knowledge Graph Completion with Mixed Geometry Tensor Factorization

Viacheslav Yusupov
HSE University

Maxim Rakhuba
HSE University

Evgeny Frolov
AIRI
HSE University

Abstract

In this paper, we propose a new geometric approach for knowledge graph completion via low rank tensor approximation. We augment a pretrained and well-established Euclidean model based on a Tucker tensor decomposition with a novel hyperbolic interaction term. This correction enables more nuanced capturing of distributional properties in data better aligned with real-world knowledge graphs. By combining two geometries together, our approach improves expressivity of the resulting model achieving new state-of-the-art link prediction accuracy with a significantly lower number of parameters compared to the previous Euclidean and hyperbolic models.

1 INTRODUCTION

Most of the information in the world can be expressed in terms of entities and the relationships between them. This information is effectively represented in the form of a knowledge graph (d’Amato, 2021; C. Peng et al., 2023), which serves as a repository for storing various forms of relational data with their interconnections. Particular examples include storing user profiles on social networking platforms (K. Xu et al., 2018), organizing Internet resources and the links between them, constructing knowledge bases that capture user preferences to enhance the functionality of recommender systems (H. Wang et al., 2019; Guo et al., 2020). With the recent emergence of large language models (LLM), knowledge graphs have become an essential tool for improving the consistency and

trustworthiness of linguistic tasks. Among notable examples of their application are fact checking (Pan et al., 2024), hallucinations mitigation (Agrawal et al., 2023), retrieval-augmented generation (P. Lewis et al., 2020), and generation of corpus for LLM pretraining (Agarwal et al., 2021). This utilization underscores the versatility and utility of knowledge graphs in managing complex datasets and facilitating the manipulation of interconnected information in various domains and downstream tasks.

On the other hand, knowledge graphs may present an incomplete view of the world. Relations can evolve and change over time, be subject to errors, processing limitations, and gaps in available information. Therefore, the tasks of restoring missing links or predicting the new ones are particularly important to maintain the accuracy and relevance of knowledge graphs. Algorithms designed to address this challenge can significantly reduce the manual labor involved in updating information and ensure consistent representation of the most current and up-to-date knowledge. In this work, we aim to design a new efficient graph completion algorithm that will *capture inherent structural properties of the underlying data to improve the link prediction quality*.

The key challenge of this task lies in the choice of proper architectural components of the solution suitable for handling a non-trivial nature of data. We start by noticing that knowledge graphs exhibit a strictly non-uniform arrangement of interconnections. Some nodes on the graph are present more frequently than others, which renders a hierarchical structure often reproducing the features of a power law-like distribution. It places knowledge graphs into the so called “complex network” category. In the seminal work on the theoretical foundations of complex networks analysis (Krioukov et al., 2010), it was shown that hyperbolic geometries are especially suitable for modelling this type of structures. In these geometries, as one moves further away from the origin, the distances between points and the area grow exponentially in contrast

to the Euclidean geometry with its respective linear and quadratic growth. Due to this effect, hyperbolic embeddings have a higher expressive ability in terms of capturing hierarchical relations within data. Consequent studies have proved the practicality of such geometric approach in various domains such as natural language processing (Maximillian Nickel and Kiela, 2017), computer vision (Ermolov et al., 2022) and recommender systems (Mirvakhabova et al., 2020).

Despite the promising results, in our experiments, we observe that real-world knowledge graphs may not consistently align with the assumption of strict hierarchical internal structure and may only partially follow a power-law distribution (see Figure 1). More specifically, the characteristics of relations in a knowledge graph or even data preprocessing may render connectivity components consisting of vertices with a large number of links almost uniformly distributed between them. We will call these vertices “active”. As we demonstrate in our work, *having a large number of active vertices violates hierarchy and disrupts the total compatibility of hyperbolic geometry with real data* (see Figure 2). Therefore, in this paper, *we aim to find a balanced solution using different geometry types* so that the weakness of each geometry is compensated. We introduce a hybrid mixed geometry model that combines the strengths of both Euclidean and hyperbolic geometry within a single architecture. In this model, the Euclidean component serves the goal of properly extracting information from the active vertices, while the hyperbolic component with embeddings from a hyperbolic space captures additional information from vertices in the graph that are distributed according to a power-law.

Overall, our contributions can be summarized as follows:

- We introduce a new mixed-geometry tensor factorisation (MIG-TF) model that combines Tucker decomposition defined in the Euclidean space with a new low-parametric hyperbolic ternary interaction term.
- We highlight intricacies of applying geometric approach to real-world knowledge graphs and demonstrate the associated with it limitations of using single-geometry modelling.
- The proposed combined approach significantly reduces the number of model parameters compared to state-of-the-art methods. It does so without sacrificing expressive power and achieves more accurate results in most of the common benchmarks.

The rest of the paper is organized as follows. In Section 2, we review most relevant Euclidean and hyper-

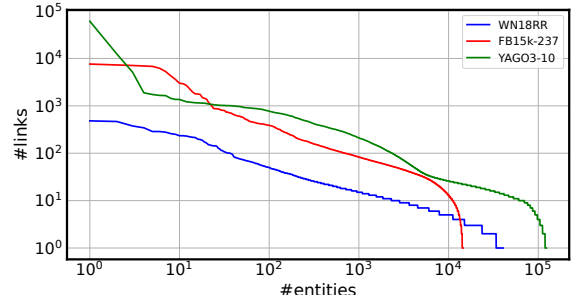


Figure 1: Links distribution on three benchmark knowledge graphs considered in this work. The FB15k-237 dataset violates the power law the most, which is indicated by the longest plateau in the left-most part of the curve. Correspondingly, the top-performing model on this dataset among previous state-of-the-art turns out to be Euclidean. In contrast, our MIG-TF approach outperforms both Euclidean and hyperbolic models, see Table 3.

bolic models. We introduce our parameter-efficient hyperbolic interaction model tetrahedron pooling tensor factorization (TPTF) along with the basics of hyperbolic geometry in Section 3.2. Section 3.3 is devoted to mixed geometry modelling MIG-TF, where we present the mixture of the Euclidean Tucker and the hyperbolic TPTF models. Finally, Section 4 contains the results of numerical experiments, where we showcase the performance of our model and also investigate the impact of using a hyperbolic correction. The anonymized code for reproducing our results is publicly available¹.

2 RELATED WORK

Knowledge graphs represent facts in the triplet form (e_s, r, e_o) , where e_s and e_o denote a pair of subject and objects entities and r denotes the corresponding relation between them. This representation can be naturally expressed in the third-order binary tensor form where a non-zero entry at position (i, j, k) encodes an existing (e_i, r_k, e_j) triplet of the knowledge graph. To identify connections in knowledge graphs that have been incorrectly specified in the interaction tensor, various tensor factorization-based models are utilized.

For instance, the *SimplE* (Kazemi and Poole, 2018) and *Complex* (Trouillon et al., 2016) models employ Canonical decomposition (CP) (Hitchcock, 1927) with embeddings in the real or complex Euclidean space. The *PITF* (Rendle and Schmidt-Thieme, 2010) model utilizes Pairwise Interaction Tensor Factorisation in

¹<https://github.com/hse-cs/MIGTF>

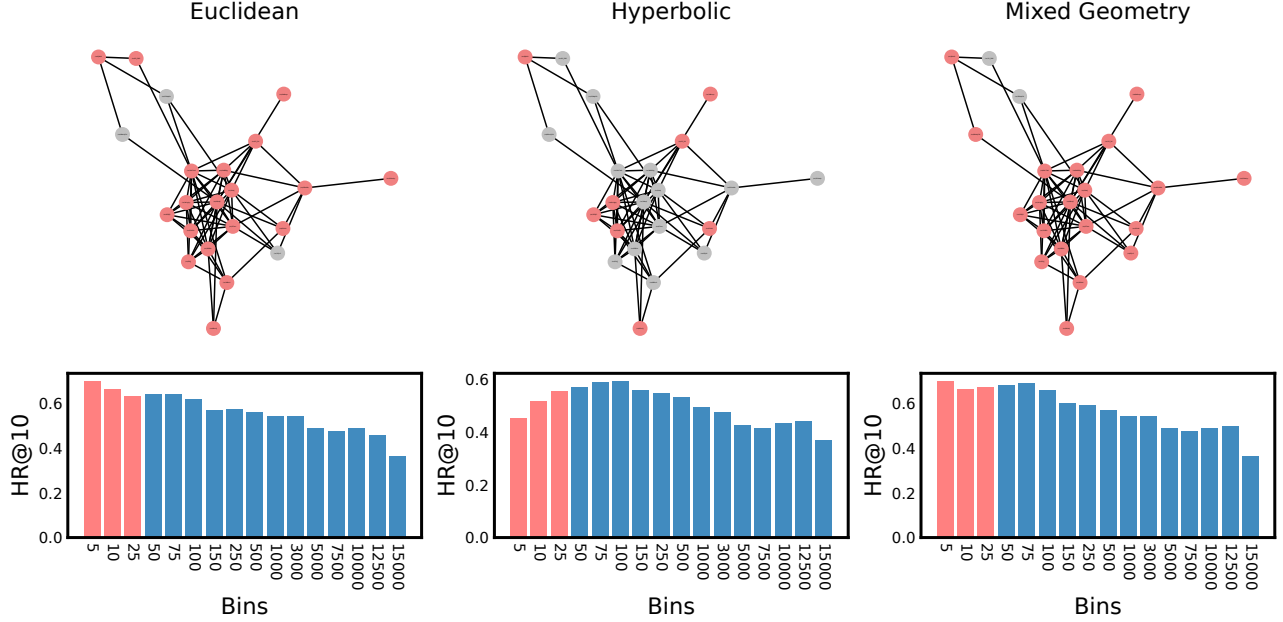


Figure 2: The graphs at the top row demonstrate the top 20 vertices in terms of their number of connections in the FB15k-237 knowledge graph. The pink marks indicate vertices that were predicted in more than 50% of cases by both the Euclidean (left), hyperbolic (center) and mixed geometry (right) models. The bar charts illustrate the hit rate (HR@10) of the predictions for links between vertices from each group. Vertices are arranged in descending order based on the number of incoming links. As can be observed, due to the high number of relations among active vertices, the hyperbolic model (TPTF) significantly underperforms in predicting relations between active vertices compared to the Euclidean model (Tucker). However, the combination of Euclidean and hyperbolic models outperforms both of them.

the Euclidean space, which splits the standard ternary interaction term of CP into a linear sum of independent interactions between entity-entity and entity-relation pairs. Some other models use distances between embeddings of entities and relations in the vector space. For instance, *TransE* (Bordes et al., 2013a) uses Euclidean vector representations of relations, which are used as connection vectors between Euclidean embeddings of entities. The further development of this approach named *Quate* (Shuai Zhang et al., 2019) represents embeddings as rotations in the hypercomplex space.

Various artificial neural networks-based approaches were also proposed to solve the link prediction task. For instance, the models like *ConvE* and *ConvKB* (Dettmers et al., 2018; Nguyen et al., 2017) utilize convolutional neural networks to work with knowledge graphs. The *R-GCN* (Schlichtkrull et al., 2018) model proposes a modification of graph-convolutional neural networks *GCN* (Kipf and Welling, 2016) to adapt graphs convolutions to relational data. Models, such as *GAAT* (R. Wang et al., 2019) or *KBGAT* (Nathani et al., 2019), use the attention mechanisms on graphs.

Some models rely on the *shared-factors paradigm*, which utilizes a special structure of the problem based on the fact that objects and subjects belong to the same space. While making the problem inherently non-linear, this approach significantly reduces the number of learned parameters of models and is better suited for predicting symmetric links (Hospedales, 2019) in knowledge graphs. For instance, the *MEKER* (Chekalina et al., 2022) model employs CP decomposition with low-rank shared factors, and *RESCAL* (Maximilian Nickel, Tresp, Kriegel, et al., 2011) utilizes *DEDICOM* (Harshman et al., 1982) factorisation decomposing each tensor slice along the relation axis. The *Tucker* (Hospedales, 2019) and *R-Tucker* (Peshekhonov, Arzhantsev, and Rakhuba, 2024) models employ shared-factors Tucker decomposition (Tucker, 1966) to represent knowledge graph entities and relations and supports the prediction of symmetric as well as asymmetric links in knowledge graphs.

Recent advances in understanding of hyperbolic geometry applications has lead to the development of hyperbolic models for knowledge graph completion as well. According to (Krioukov et al., 2010), the structural

properties of hyperbolic spaces make them plausible for analyzing a special family of non-uniformly distributed data, which knowledge graphs also belong to. For instance, the *RotH* and *RefH* (Re, 2020) propose specific rotation and reflection transformations in a hyperbolic space to improve the expressive power of the learned embeddings. Even though these model show remarkable improvements on some knowledge graphs, *there are still some cases where Euclidean models outperform hyperbolic ones*. We attribute this observation to the violation of some basic distributional properties required for full compatibility with hyperbolic geometry. It motivates us to develop a hybrid approach similarly to (Gu et al., 2019; Seth and Sharaff, 2022; Kong, 2019) that takes the best from both geometries and therefore compensates for their shortcomings in analyzing real-world data. Similar idea was utilized in models *M²GNN* (S. Wang et al., 2021) generalizing graph neural network for multi-curvature and *GIE* (Cao et al., 2022) introducing message passing and attention algorithms on hyperbolic and hyperspherical spaces. In comparison to previous methods, our MIG-TF approach uses less complex and more stable operations (Section 3.4). Additionally, it implies learning only low-parametric hyperbolic addition to a pretrained model instead of learning the whole model as in *RotH*, *M²GNN* and *GIE*. Further in the text, we provide the detailed description of our approach.

3 MIXED GEOMETRY TENSOR FACTORIZATION

In this section, we introduce our new parameter-efficient hyperbolic interaction term TPTF, and the new mixed geometry model MIG-TF, where the hyperbolic term is used as a correction to the Euclidean model.

3.1 Hyperbolic geometry and Lorentz Distance

We start with basic notation and concepts from hyperbolic geometry. There are several models of hyperbolic geometry, among which Poincare Ball, Klein, and Lorentz model are frequently used in various machine learning approaches. All hyperbolic models are isomorphic – they describe the same geometry and can be interchangeably utilized by means of the corresponding transformations (Ungar, 2005). In our work, we use the Lorentz model of hyperbolic geometry. In the following, we only introduce the basics of Lorentz geometry and refer the reader to (Beem, 2017) for more details.

Let us define the Lorentz inner product for vectors $x, y \in \mathbb{R}^{n+1}$, which is used for measuring distance be-

tween hyperbolic embeddings:

$$\langle x, y \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^n x_i y_i,$$

where

$$x_0 = \sqrt{\beta + \sum_{i=1}^n x_i^2}.$$

The induced norm of a vector $x \in \mathbb{R}^{n+1}$ in Lorentz geometry is defined as $\|x\|_{\mathcal{L}}^2 = \langle x, x \rangle_{\mathcal{L}} = -\beta$. The corresponding n -dimensional space $\mathcal{H}^{n,\beta} \subset \mathbb{R}^{n+1}$ is called Hyperboloid and defined as follows:

$$\mathcal{H}^{n,\beta} = \{x \in \mathbb{R}^{n+1} \mid \|x\|_{\mathcal{L}}^2 = -\beta, \beta \geq 0\}.$$

The origin vector of the hyperboloid $\mathcal{H}^{n,\beta}$ equals to $\mathbf{0} = (\beta, 0, \dots, 0) \in \mathbb{R}^{n+1}$. The inner product of $\mathbf{0}$ and x is, hence, $\langle \mathbf{0}, x \rangle_{\mathcal{L}} = -\beta x_0$.

The associated geodesic distance is defined as

$$d_l(x, y) = \text{arccosh}(-\langle x, y \rangle_{\mathcal{L}}).$$

Similarly to (C. Xu and Wu, 2020), we introduce the square Lorentz distance between $x, y \in \mathcal{H}^n$ is defined as

$$d_{\mathcal{L}}^2(x, y) = \|x - y\|_{\mathcal{L}}^2 = -2 - 2\langle x, y \rangle_{\mathcal{L}}. \quad (1)$$

This distance fulfills all of the Euclidean distance axioms, except for the triangle inequality:

$$d(x, z) \leq d(x, y) + d(y, z). \quad (2)$$

Nevertheless, it does not hold in general in the hyperbolic setting. The idea to benefit from this fact appears, for instance, in a two-dimensional hyperbolic model *LorentzFM* (C. Xu and Wu, 2020). Inequality violation in this model implies that two entities are far away and, hence, become irrelevant to each other. Generalizing the triangle inequality to the three-dimensional case is not a straightforward task, which we address in the following section.

3.2 Tetrahedron Pooling Tensor Factorization

Solving the knowledge graph link prediction problem requires learning ternary interactions in the form of entity-relation-entity triplets. This can be naturally modelled through tensor representation in the Euclidean space. However, unlike the 2D case, generalizing a representation of such third-order interactions to the hyperbolic space is not straightforward. To the best of our knowledge, there is no unified approach.

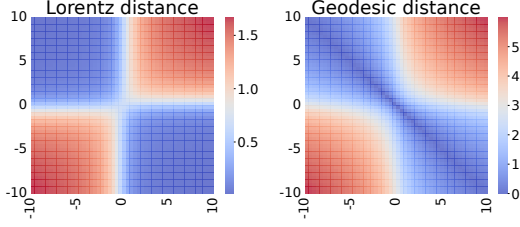


Figure 3: Left score function landscape corresponds to our score function (5), whilst right one corresponds to (4). As seen from the plots the landscapes of score functions for different distances are similar. For more score function landscapes see Appendix.

We propose to modify (2) to capture ternary interactions. In particular, we utilize the so-called tetrahedron inequality (Itoh, Rouyer, and Vilcu, 2021): for the points u, v, t, o in the Euclidean space, it holds

$$d(u, v) + d(o, t) \leq d(u, t) + d(v, t) + d(o, u) + d(o, v). \quad (3)$$

In the hyperbolic context, this inequality does not universally apply. However, as we explore further, the regions where this inequality is violated can be advantageous for our objectives. Consequently, we can naturally introduce the following score function:

$$S_G = d_l(u, v) + d_l(\mathbf{0}, t) - d_l(u, t) - d_l(v, t) - d_l(\mathbf{0}, u) - d_l(\mathbf{0}, v), \quad (4)$$

where $u, v \in \mathbb{R}^{d_l}$ are entity embeddings and $t \in \mathbb{R}^{d_l}$ is the relation embedding. Note that this score function is negative when inequality (3) holds and is positive in the opposite case. Unfortunately, the score function (4) involves arccosh , which is not differentiable everywhere and, hence, may lead to difficulties for gradient-based optimization methods.

Similarly to *LorentzFM*, (4) is replaced with the following “smoothed” score function that mimics the desired behaviour:

$$S_H(u, v, t) = \frac{\frac{1}{2} \begin{pmatrix} d_{\mathcal{L}}^2(u, v) + d_{\mathcal{L}}^2(\mathbf{0}, t) - d_{\mathcal{L}}^2(u, t) \\ -d_{\mathcal{L}}^2(t, v) - d_{\mathcal{L}}^2(\mathbf{0}, u) - d_{\mathcal{L}}^2(\mathbf{0}, v) \end{pmatrix}}{\langle \mathbf{0}, v \rangle_{\mathcal{L}} \langle \mathbf{0}, t \rangle_{\mathcal{L}} + \langle \mathbf{0}, u \rangle_{\mathcal{L}} \langle \mathbf{0}, v \rangle_{\mathcal{L}} + \langle \mathbf{0}, t \rangle_{\mathcal{L}} \langle \mathbf{0}, v \rangle_{\mathcal{L}}}, \quad (5)$$

see Figure 3 for comparison of the two score functions.

Note also that we may rewrite S_H as:

$$S_H = \frac{(2 + t_0 - u_0 - v_0) - (\langle u, v \rangle_{\mathcal{L}} - \langle u, t \rangle_{\mathcal{L}} - \langle t, v \rangle_{\mathcal{L}})}{u_0 v_0 + u_0 t_0 + t_0 v_0}.$$

The score function yields a positive value when the points corresponding to linked entities are in close proximity and a negative value when they are distant

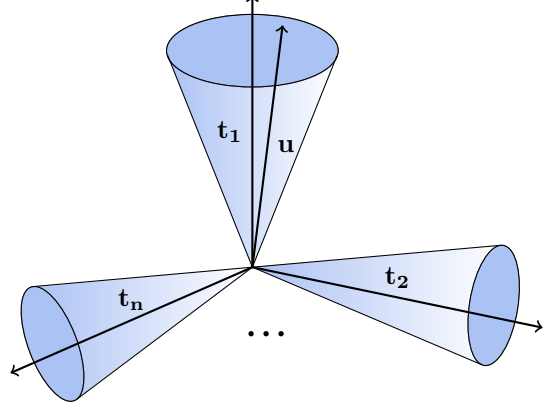


Figure 4: Each embedding of a relation (t_1, \dots, t_n) defines a cone that encompasses the embeddings of the entities associated with that relation.

(see Appendix). Additionally, we note that the terms $d_{\mathcal{L}}^2(u, t)$ and $d_{\mathcal{L}}^2(t, v)$, which account for interactions between relations and entities, effectively draw entities u and v closer to the relation t . Consequently, the trained model exhibits a distinct behavior: entities tend to cluster around well-separated relations, as illustrated in Figure 4.

The model’s predictions p for entity u and relation t can be obtained with $p = \{p_j\}_{j=1}^{n_e}$:

$$p = \sigma(S_H(u, V, t)),$$

where $V \in \mathbb{R}^{d_l \times n_e}$ is the matrix of all entity embeddings and n_e – the number of entities. In TPTF model, we use the binary cross-entropy (BCE) loss function:

$$l_{BCE}(y_i, p_j) = -y_i \log(p_j) - (1 - y_i) \log(1 - p_j),$$

where $y = \{y_i\}_{i=1}^{n_e}$ – binary label vector of the ground truth. Similarly to (Law et al., 2019), we optimize u , v and t embeddings in the Euclidean space with the *AdamW* (Loshchilov and Hutter, 2019) optimizer and then map them to the hyperboloid as follows: $v \in \mathbb{R}^n$ maps to $[\sqrt{\beta + \|v\|_2^2}, v] \in \mathbb{R}^{n+1}$.

3.3 Mixed Geometry Tensor Factorisation

To improve the link prediction in knowledge graphs we introduce a shared-factor mixed geometry model combining Euclidean and hyperbolic models (see Figure 5). Due to the complexity of the knowledge graph, some vertices are better described with hyperbolic embeddings whilst others are with the Euclidean ones. Therefore with a combination of hyperbolic and Euclidean embeddings, significantly more information on the structure of the knowledge graph is saved with Euclidean and hyperbolic vector representations together. For the hyperbolic model, we used TPTF

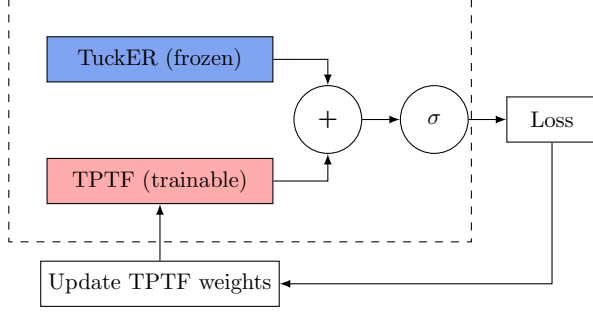


Figure 5: The proposed MIG-TF model architecture.

model (5), and for Euclidean one we used TuckER (Hospedales, 2019). The score function of the Euclidean model is defined as:

$$S_E(G, u, v, t) = \sum_{\alpha, \beta, \gamma=1}^{d_1, d_1, d_2} G_{\alpha\beta\gamma} u_\alpha v_\beta t_\gamma, \quad (6)$$

where $u, v \in \mathbb{R}^{d_1}$ – embeddings of entities, $t \in \mathbb{R}^{d_2}$ – embedding of relations and $G \in \mathbb{R}^{d_1 \times d_1 \times d_2}$ – tensor of learnable parameters. Here, d_1 and d_2 are the ranks of the Tucker decomposition.

The score function of our mixed-geometry model MIG-TF is the sum of score functions of the Lorentzian and Euclidean models:

$$(S_{MIG-TF})_i = (S_E)_i + (S_H)_i, \quad (7)$$

where $(S_E)_i = S_E(G, u, V_i, t)$ defined in (6) is an Euclidean interaction term, $(S_H)_i = S_H(u^\mathcal{L}, V_i^\mathcal{L}, t^\mathcal{L})$ is a hyperbolic interaction term (5). The vectors u and t – Euclidean entity and relation embeddings, $u^\mathcal{L}$ and $t^\mathcal{L}$ – hyperbolic embeddings, V_i and $V_i^\mathcal{L}$ are i -th rows of matrices V and $V^\mathcal{L}$ of n_e Euclidean and hyperbolic entity embeddings, respectively, and $G \in \mathbb{R}^{d_1 \times d_1 \times d_2}$ – 3-D tensor of learnable parameters. In MIG-TF model, we utilize pretrained TuckER model (6) and optimize the hyperbolic term parameters (5) of the score function (7) to minimize the BCE loss:

$$\mathcal{L}_{MIG-TF} = \frac{1}{n_e} \sum_{i=1}^{n_e} l_{BCE}(a_i, \sigma((S_{MIG-TF})_i)),$$

where a_i is an i -th element of interaction vector a for entity and relation with embeddings u and t , respectively. We optimize hyperbolic embeddings as in Section 3.2. To prevent model overfitting, we employ the dropout layer (Salakhutdinov, 2012) in both Euclidean and hyperbolic interaction terms.

3.4 Mixed-geometry models comparison

The next subsection is devoted to comparing our approach with other mixed-geometry methods: M^2GNN

(S. Wang et al., 2021) and GIE (Cao et al., 2022).

The translational distance (Siheng Zhang, Sun, and W. Zhang, 2020) model M^2GNN generalizes graph neural networks for multi-relational setting on knowledge graphs and modified it for multi-curvature setting capturing graph structures more effectively, especially hierarchical and cyclic ones.

The model GIE introduces the message passing and attention mechanism for the knowledge graph completion by utilizing mapping on hyperbolic and hyper-spherical spaces.

Both of the existing approaches suffers from computational complexity and numerical instability due to utilizing exponential and logarithmic mapping, which are even more numerically unstable than the standard Lorentz model operations (W. Peng et al., 2021). In contrast, MIG-TF proposes using Square Lorentz distance, a convenient replacement for Geodesic distance. Square Lorentz distance reduces hyperbolic operations, such as exponential and logarithmic mappings and arccosh, to mere inner product computations, decreases complexity of operations and increases numerical stability.

Additionally, only the low-parametric addition TPTF is learned in our MIG-TF method instead of learning the whole models in M^2GNN and GIE .

Unlike (S. Wang et al., 2021) and (Cao et al., 2022), we do not utilize spherical geometry and still our mixed geometry approach achieves the best metrics in the majority of cases. Nevertheless, our framework is sufficiently flexible to accommodate the incorporation of spherical geometry, presenting a promising direction for future research.

4 EXPERIMENTS

In experiments, we shows that mixed geometry model, combining Euclidean and hyperbolic embeddings, with low-dimensional hyperbolic ones outperforms Euclidean, Complex and hyperbolic models with significantly higher dimensions of embeddings. Additionally, we have studied the dependence of curvature of the space of hyperbolic embeddings and expressive ability of hyperbolic and mixed geometry models. Moreover, we have shown that hyperbolic addition to TuckER (Hospedales, 2019) model improves the quality of link prediction between hierarchically distributed entities.

4.1 Datasets

We evaluate our mixed geometry model for link prediction task on three standard knowledge graphs: WN18RR (Bordes et al., 2013a; Dettmers et al.,

2018), FB15k-237 (Bordes et al., 2013a; Toutanova et al., 2015) and YAGO3-10 (Mahdisoltani, Biega, and Suchanek, 2013). WN18RR is a subset of WordNet (Miller, 1995) – knowledge base containing lexical relations between words. FB15k-237 is a subset of Free-Base (Bollacker et al., 2008) – collaborative knowledge base containing general world knowledge, such as nationalities of famous people, locations of some buildings or genre of music. YAGO3-10 is a subset of YAGO3 (Suchanek, Kasneci, and Weikum, 2007) which mostly contains descriptions of people. The statistics of these knowledge graphs are shown in following table: For each knowledge graph we follow stan-

Table 1: Knowledge graphs statistics.

Dataset	Entities	Relations	Triplets
WN18RR	40,943	11	93,003
FB15k-237	14,541	237	310,116
YAGO3-10	123,182	37	1,179,040

dard data augmentation by adding inverse relations (Bordes et al., 2013b).

4.2 Baselines

We compare our models with 8 baselines: 4 models with euclidean embeddings, 2 with complex embeddings and 2 with hyperbolic embeddings:

- **Euclidean models:** *PITF* (Pairwise Interaction Tensor Factorisation) – model using PITF tensor decomposition and Euclidean embeddings (Rendle and Schmidt-Thieme, 2010). *DistMult* (Distance-based Multiplication) – model utilizing canonical tensor decomposition (Yang et al., 2014).
- **TuckER-based Euclidean models:** TuckER (Tucker decomposition) – model using Tucker tensor decomposition and Euclidean embeddings (Hospedales, 2019). *RTuckER* – TuckER model with a Riemannian method of optimization (Peshekhonov, Arzhantsev, and Rakhuba, 2024).
- **Complex models:** *Complex-N3* (Complex Embeddings) – model using real part of the canonical tensor decomposition and complex embeddings (Chen et al., 2021). *RotatE* (Relational Rotation in complex space) – model using rotations in complex space and complex embeddings (Tang, 2019).
- **Hyperbolic models:** *RotH* (Hyperbolic Rotations) and *RefH* (Hyperbolic Reflections) – attention models using rotations and reflections in

hyperbolic space and utilizing hyperbolic embeddings from Poincare Ball (Re, 2020).

- **Mixed-geometry models:** *M²GNN* (Mixed-Curvature Multi-Relational Graph Neural Network) – model using multi-curvature embeddings for translational distance and multi-relational graph neural networks. *GIE* (Geometry Interaction Knowledge Graph Embeddings) – model utilizing attention layers and embeddings from Euclidean, hyperbolic and spherical spaces.

To demonstrate the benefits of our mixed geometry approach to link prediction task we compare baselines with our hyperbolic model TPTF (5) and our mixed geometry models MIG-TF (7) and MIG-TF_{QR}. Here, MIG-TF_{QR} is mixed geometry model with orthogonalization heuristic. For more details see Appendix. Additionally, to better understand the role of hyperbolic interaction term we compare TPTF and MIG-TF models with different fixed hyperbolicities.

4.3 Evaluation metrics

At the test mode, we rank correct entities against all entities using score functions (5) for TPTF and (7) for MIG-TF. We compare models using two rank-based metrics: *MRR* (Mean Reciprocal Rank) and *HR@k*, $k \in 1, 3, 10$ (Hit Rates). *MRR* measures mean inverse rank of correct entity, whilst *HR@k* measures the share of correct triples among *Top-k* recommendations (Vasilev, 2021). We follow the standard evaluation protocol in the filtered setting: all true triples in the KG are filtered out during evaluation (Lacroix, Usunier, and Obozinski, 2018).

5 RESULTS

Our mixed geometry models achieved new state-of-the-art results on FB15k-237, YAGO3-10 and WN18RR in the majority of metrics. Moreover, our models have significantly less parameters than the best models on WN18RR and YAGO3-10. For instance, our models have 8 times less parameters than *RotH* or *RefH*. Additionally, as we expected, Lorentz addition to the TuckER model significantly increases *MRR* and *HR@k* metrics, with low - Lorentzian embeddings. We also show, that MIG-TF model is better than TuckER model in prediction of various relations, especially rare ones Figure 6.

5.1 Hyperbolicity

To study the impact of hyperbolic interaction term in hybrid model we trained TPTF and MIG-TF models with various fixed curvatures. Models with low

Table 2: Approximate number of models’ parameters on knowledge graphs *WN18RR*, *FB15k* – 237 and *YAGO3* – 10.

Models	FB15k-237	WN18RR	YAGO3-10
<i>PITF</i>	$4 \cdot 10^6$	$8 \cdot 10^6$	$25 \cdot 10^6$
<i>DistMult</i>	$4 \cdot 10^6$	$8 \cdot 10^6$	$25 \cdot 10^6$
<i>TuckER</i> $_{S_E+0 \cdot S_H}$	$4 \cdot 10^6$	$8 \cdot 10^6$	$25 \cdot 10^6$
<i>RTuckER</i> $_{S_E+0 \cdot S_H}$	$4 \cdot 10^6$	$8 \cdot 10^6$	$25 \cdot 10^6$
<i>Complex-N3</i>	$4 \cdot 10^6$	$8 \cdot 10^6$	$25 \cdot 10^6$
<i>RotatE</i>	$60 \cdot 10^6$	$60 \cdot 10^6$	$120 \cdot 10^6$
<i>RotH</i>	$40 \cdot 10^6$	$80 \cdot 10^6$	$120 \cdot 10^6$
<i>RefH</i>	$40 \cdot 10^6$	$80 \cdot 10^6$	$120 \cdot 10^6$
<i>M²GNN</i>	$12 \cdot 10^6$	$24 \cdot 10^6$	$75 \cdot 10^6$
<i>GIE</i>	$4 \cdot 10^6$	$18 \cdot 10^6$	$24 \cdot 10^6$
Our models			
<i>TPTF</i> $_{0 \cdot S_E+S_H}$	$2 \cdot 10^6$	$4 \cdot 10^6$	$12 \cdot 10^6$
<i>MIG-TF</i> $_{S_E+S_H}$	$5 \cdot 10^6$	$10 \cdot 10^6$	$31 \cdot 10^6$

curvature are similar to euclidean ones, therefore they worse predict links between hierarchically distributed entities of knowledge graph. In contrast, models with curvature $c \approx 1.5$ show the highest metrics both in mixed geometry and hyperbolic cases Figure 7.

Euclidean component in MIG-TF model embeds features of knowledge graph structure, whilst hyperbolic ones are not fully represented with embeddings. Therefore, the hyperbolic component of mixed geometry model represents only the hyperbolic features of knowledge graph that left after Euclidean component application (Ni et al., 2015; Gu et al., 2019). In contrast, *RotH* and *RefH* architectures (Re, 2020), are based on rotations and reflections in hyperbolic and the Euclidean space, respectively. In contrast, our model utilizes the hyperbolic triangle inequality, as well as uses embeddings in both Euclidean and hyperbolic spaces. Additionally, our model has less stringent requirements on the structure of the input data.

5.2 Link prediction quality

As demonstrated in Table 3, our mixed geometry approach with mixed geometries outperforms all other competitors on the two out of three datasets. It slightly underperforms its main competitor (the hyperbolic *RotH* model) on the *WN18RR* dataset on one metric. We attribute this result to the fact that the structure of this dataset is the most plausible for analysis with hyperbolic geometry. Indeed, by inspecting Figure 1, we see that this dataset is the most compatible with the ideal power law distribution, it contains the minimal amount of active nodes and exhibits almost linear dependency over the majority of entities. This result aligns well with our initial design that is targeted towards mixed hierarchical and non-hierarchical structures and therefore is may not take

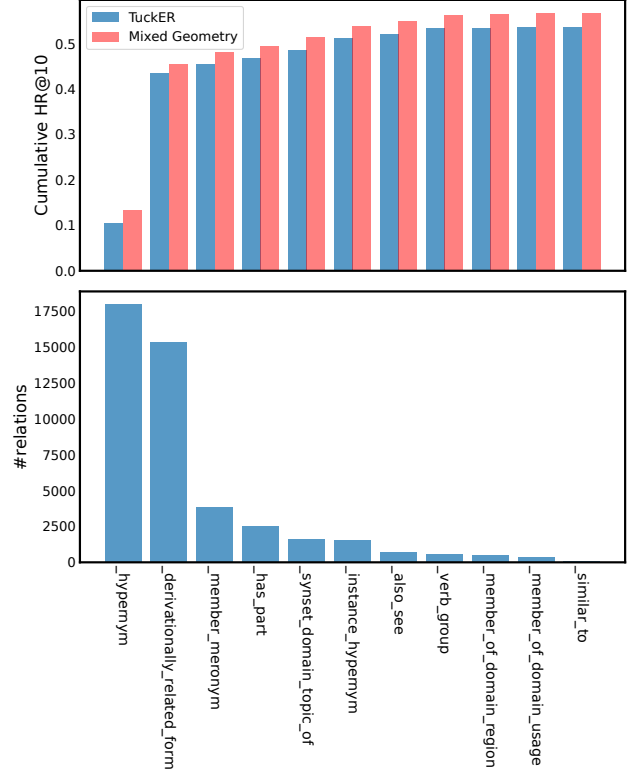


Figure 6: The top bar chart demonstrates the cumulative $HR@10$ metrics in the *WN18RR* knowledge graph with respect to all types of relations ordered by their occurrence frequency. It is seen that Mixed Geometry model (red) has higher cumulative metrics in comparison to TuckER (blue). In addition, Mixed Geometry has significant increase in $HR@10$ metric on rare relations. For instance, on “member of domain region” +140%, “member of domain usage” +48% and “member meronym” +36%. The bottom bar chart illustrates the number of links with each relation in knowledge graph.

full advantage of more homogeneous data organization. A possible remedy to this effect would be changing Euclidean component of MIG-TF model to more powerful model, such as Euclidean *RotE* (Re, 2020) or complex *Complex-N3* (Chen et al., 2021). We leave investigation of this approach for future work.

On the other two datasets our approach dominates its competitors, which aligns well with our assumptions on the structural complexities of these datasets that prevent purely hyperbolic or purely Euclidean model to learn the best representations. Our approach not only outperforms other models but also requires a lesser amount of parameters, which indicates the higher representation capacity of the mixed geometry embeddings. By consulting to Figure 1, one can see

Table 3: Metrics on knowledge graphs: $HR@k$ and MRR metrics of models on knowledge graphs WN18RR, FB12k-237 and YAGO3-10. **Bold** means the best metric on the knowledge graph and underlined – the second best metric. For each knowledge graph we follow standard data augmentation by adding inverse relations (Bordes et al., 2013b).

Models	FB15k-237				WN18RR				YAGO3-10			
	HR@10	HR@3	HR@1	MRR	HR@10	HR@3	HR@1	MRR	HR@10	HR@3	HR@1	MRR
<i>PITF</i>	0.459	0.347	0.186	0.284	0.403	0.384	0.392	0.296	0.610	0.494	0.350	0.453
<i>DistMult</i>	0.419	0.263	0.155	0.241	0.490	0.440	0.390	0.430	0.540	0.380	0.240	0.340
<i>TuckER</i> $_{S_E+0 \cdot S_H}$	0.544	<u>0.394</u>	<u>0.266</u>	0.358	0.526	0.482	0.443	0.470	0.681	0.579	0.466	0.544
<i>RTuckER</i> $_{S_E+0 \cdot S_H}$	0.505	0.387	0.237	0.326	0.546	0.495	0.446	0.479	–	–	–	–
<i>ComplEx</i> -N3	0.547	0.392	0.264	0.357	0.572	0.495	0.435	0.480	0.701	0.609	0.498	0.569
<i>RotatE</i>	0.533	0.375	0.245	0.338	0.571	0.492	0.428	0.476	0.670	0.550	0.402	0.495
<i>RotH</i>	0.535	0.380	0.246	0.344	0.586	0.514	0.449	<u>0.496</u>	0.706	0.612	0.495	0.573
<i>RefH</i>	0.536	0.383	0.252	0.346	0.568	0.485	0.404	0.461	0.711	0.619	<u>0.502</u>	0.576
<i>M²GNN</i>	0.565	0.398	<u>0.275</u>	0.362	0.572	0.498	0.444	0.485	0.702	0.605	0.478	0.573
<i>GIE</i>	0.552	<u>0.401</u>	0.271	0.362	<u>0.575</u>	0.505	0.452	0.491	0.709	0.618	0.505	<u>0.579</u>
Our models												
<i>TPTF</i> $_{0 \cdot S_E+S_H}$	0.504	0.321	0.186	0.238	0.489	0.378	0.252	0.314	0.665	0.521	0.383	0.481
<i>MIG-TF</i> $_{S_E+S_H}$	<u>0.554</u>	0.402	0.277	0.367	0.561	<u>0.507</u>	<u>0.450</u>	<u>0.496</u>	<u>0.717</u>	<u>0.621</u>	<u>0.502</u>	<u>0.579</u>
<i>MIG-TF</i> $_{Q_R, S_E+S_H}$	0.553	0.402	0.277	<u>0.366</u>	0.574	0.514	0.452	0.499	0.720	0.622	<u>0.502</u>	0.580

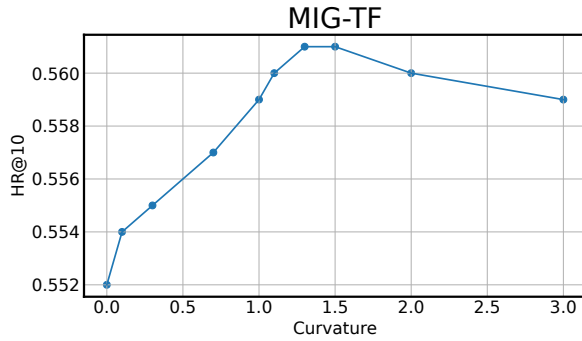


Figure 7: The graph presents the dependence of $HR@10$ metric from curvature of Hyperbolic space β for MIG-TF model on WN18RR dataset.

that the most prominent results are obtained on the *FB15k – 237* dataset exhibiting the highest degree of violation of the power law distribution. Remarkably, the otherwise top-performing hyperbolic models underperform the best Euclidean model on this dataset. This signifies the detrimental effect of the structural non-homogeneity on single-geometry models and highlight the advantages of our mixed geometry approach.

6 CONCLUSION

The use of the proposed mixed geometry model MIG-TF combining the Euclidean and hyperbolic embedding techniques can significantly improve the quality of link predictions in knowledge graphs with non-homogeneous structure. It provides a significant advantage over state-of-the-art single-geometry models, such as the Euclidean TuckER and the hyperbolic *RotH* and *RefH*. This effect can be attributed to the

fact that both Euclidean and Hyperbolic embeddings capture different aspects of relationships between entities in a knowledge graph. By combining these two approaches, mixed geometry models can capture a broader range of relationships, leading to more accurate predictions. We also show that hybrid models are particularly well-suited for addressing the challenge of link prediction in knowledge graphs under limited computing resources. Due to the geometry-driven improved representation capacity, our proposed approach leads to fewer parameters compared to complex or attention-based models such as *RotatE* and *RotH*. Additionally, our model requires training only the minority of its parameter in comparison to the other mixed geometry models such as *GIE* and *M²GNN*. This facts make MIG-TF approach more efficient and scalable.

Acknowledgments

The work of Viacheslav Yusupov and Maxim Rakhuba was prepared within the framework of the HSE University Basic Research Program. The calculations were performed in part through the computational resources of HPC facilities at HSE University (Kostenetskiy, Chulkevich, and Kozyrev, 2021).

References

Agarwal, Oshin et al. (June 2021). “Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 3554–3565.

- Agrawal, Garima et al. (2023). “Can knowledge graphs reduce hallucinations in LLMs?: A survey”. In: *arXiv preprint arXiv:2311.07914*.
- Beem, John K (2017). *Global lorentzian geometry*. Routledge.
- Bollacker, Kurt et al. (2008). “Freebase: a collaboratively created graph database for structuring human knowledge”. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250.
- Bordes, Antoine et al. (2013a). “Translating embeddings for modeling multi-relational data”. In: *Advances in neural information processing systems* 26.
- (2013b). “Translating embeddings for modeling multi-relational data”. In: *Advances in neural information processing systems* 26.
- Cao, Zongsheng et al. (2022). “Geometry interaction knowledge graph embeddings”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 5, pp. 5521–5529.
- Chekalina, Viktoriia et al. (May 2022). “MEKER: Memory Efficient Knowledge Embedding Representation for Link Prediction and Question Answering”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Dublin, Ireland: Association for Computational Linguistics, pp. 355–365.
- Chen, Yihong et al. (2021). “Relation Prediction as an Auxiliary Training Objective for Improving Multi-Relational Graph Representations”. In: *3rd Conference on Automated Knowledge Base Construction*.
- d’Amato, Aidan Hogan Eva Blomqvist Michael Cochez Claudia (2021). “Knowledge Graphs”. In: *ACM*.
- Dettmers, Tim et al. (2018). “Convolutional 2d knowledge graph embeddings”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1.
- Ermolov, Aleksandr et al. (2022). “Hyperbolic vision transformers: Combining improvements in metric learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7409–7419.
- Gu, Albert et al. (2019). “Learning mixed-curvature representations in products of model spaces”. In: *International conference on learning representations*. Vol. 5.
- Guo, Qingyu et al. (2020). “A survey on knowledge graph-based recommender systems”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.8, pp. 3549–3568.
- Harshman, Richard A et al. (1982). “A model for the analysis of asymmetric data in marketing research”. In: *Marketing Science* 1.2, pp. 205–242.
- Hitchcock, Frank L (1927). “The expression of a tensor or a polyadic as a sum of products”. In: *Journal of Mathematics and Physics* 6.1-4, pp. 164–189.
- Hospedales, Ivana Balažević Carl Allen Timothy M. (2019). “TuckER: Tensor Factorization for Knowledge Graph Completion”. In: *ACL*.
- Itoh, Jin-ichi, Joël Rouyer, and Costin Vilcu (2021). “Some inequalities for tetrahedra”. In: *Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry* 62, pp. 705–715.
- Kazemi, Seyed Mehran and David Poole (2018). “Simple embedding for link prediction in knowledge graphs”. In: *Advances in neural information processing systems* 31.
- Kipf, Thomas N and Max Welling (2016). “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907*.
- Kong, Rui Chen Qingyi Hua Yan-Shuo Chang Bo Wang Lei Zhang Xiangjie (2019). “A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks”. In: *IEEE*.
- Kostenetskiy, PS, RA Chulkevich, and VI Kozyrev (2021). “HPC resources of the higher school of economics”. In: *Journal of Physics: Conference Series*. Vol. 1740. 1. IOP Publishing, p. 012050.
- Krioukov, Dmitri et al. (2010). “Hyperbolic geometry of complex networks”. In: *Physical Review E* 82.3, p. 036106.
- Lacroix, Timothée, Nicolas Usunier, and Guillaume Obozinski (2018). “Canonical tensor decomposition for knowledge base completion”. In: *International Conference on Machine Learning*. PMLR, pp. 2863–2872.
- Law, Marc et al. (2019). “Lorentzian distance learning for hyperbolic representations”. In: *International Conference on Machine Learning*. PMLR, pp. 3672–3681.
- Lewis, Patrick et al. (2020). “Retrieval-augmented generation for knowledge-intensive nlp tasks”. In: *Advances in Neural Information Processing Systems* 33, pp. 9459–9474.
- Loshchilov, Ilya and Frank Hutter (2019). “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*.
- Mahdisoltani, Farzaneh, Joanna Biega, and Fabian M Suchanek (2013). “Yago3: A knowledge base from multilingual wikipedias”. In: *CIDR*.
- Miller, George A (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41.

- Mirvakhabova, Leyla et al. (2020). “Performance of hyperbolic geometry models on top-N recommendation tasks”. In: *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 527–532.
- Nathani, Deepak et al. (2019). “Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs”. In: *Annual Meeting of the Association for Computational Linguistics*.
- Nguyen, Dai Quoc et al. (2017). “A novel embedding model for knowledge base completion based on convolutional neural network”. In: *arXiv preprint arXiv:1712.02121*.
- Ni, Chien-Chun et al. (2015). “Ricci curvature of the internet topology”. In: *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, pp. 2758–2766.
- Nickel, Maximilian, Volker Tresp, Hans-Peter Kriegel, et al. (2011). “A three-way model for collective learning on multi-relational data.” In: *ICML*. Vol. 11. 10.5555, pp. 3104482–3104584.
- Nickel, Maximilian and Douwe Kiela (2017). “Poincaré embeddings for learning hierarchical representations”. In: *Advances in neural information processing systems* 30.
- Pan, Shirui et al. (2024). “Unifying large language models and knowledge graphs: A roadmap”. In: *IEEE Transactions on Knowledge and Data Engineering*.
- Peng, Ciyuan et al. (2023). “Knowledge graphs: Opportunities and challenges”. In: *Artificial Intelligence Review* 56.11, pp. 13071–13102.
- Peng, Wei et al. (2021). “Hyperbolic deep neural networks: A survey”. In: *IEEE Transactions on pattern analysis and machine intelligence* 44.12, pp. 10023–10044.
- Peshekhonov, Ivan, Aleksey Arzhantsev, and Maxim Rakhuba (2024). “Training a Tucker Model With Shared Factors: a Riemannian Optimization Approach”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 3304–3312.
- Re, Ines Chami Adva Wolf Da-Cheng Juan Frederic Sala Sujith Ravi Christopher (2020). “Low-Dimensional Hyperbolic Knowledge Graph Embeddings”. In: *ACL*.
- Rendle, Steffen and Lars Schmidt-Thieme (2010). “Pairwise interaction tensor factorization for personalized tag recommendation”. In: *Proceedings of the third ACM international conference on Web search and data mining*, pp. 81–90.
- Salakhutdinov, Geoffrey E. Hinton Nitish Srivastava Alex Krizhevsky Ilya Sutskever Ruslan R. (2012). “Improving neural networks by preventing co-adaptation of feature detectors”. In: *CoRR*.
- Schlichtkrull, Michael et al. (2018). “Modeling relational data with graph convolutional networks”. In: *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer, pp. 593–607.
- Seth, Rakhi and Aakanksha Sharriff (2022). “A comparative overview of hybrid recommender systems: Review, challenges, and prospects”. In: *Data mining and machine learning applications*, pp. 57–98.
- Suchanek, Fabian M, Gjergji Kasneci, and Gerhard Weikum (2007). “Yago: a core of semantic knowledge”. In: *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706.
- Tang, Zhiqing Sun Zhi-Hong Deng Jian-Yun Nie Jian (2019). “Rotate: Knowledge graph embedding by relational rotation in complex space”. In: *ICLR*.
- Toutanova, Kristina et al. (2015). “Representing text for joint embedding of text and knowledge bases”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1499–1509.
- Trouillon, Théo et al. (2016). “Complex embeddings for simple link prediction”. In: *International conference on machine learning*. PMLR, pp. 2071–2080.
- Tucker, Ledyard R (1966). “Some mathematical notes on three-mode factor analysis”. In: *Psychometrika* 31.3, pp. 279–311.
- Ungar, Abraham A (2005). *Analytic hyperbolic geometry: Mathematical foundations and applications*. World Scientific.
- Vasilev, Yan-Martin Tamm Rinchin Damdinov Alexey (2021). “Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently?” In: *ACM*.
- Wang, Hongwei et al. (2019). “Knowledge graph convolutional networks for recommender systems”. In: *The world wide web conference*, pp. 3307–3313.
- Wang, Rui et al. (2019). “Knowledge graph embedding via graph attenuated attention networks”. In: *IEEE access* 8, pp. 5212–5224.
- Wang, Shen et al. (2021). “Mixed-curvature multi-relational graph neural network for knowledge graph completion”. In: *Proceedings of the web conference 2021*, pp. 1761–1771.
- Xu, Canran and Ming Wu (2020). “Learning feature interactions with lorentzian factorization machine”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 6470–6477.
- Xu, Keyulu et al. (2018). “Representation learning on graphs with jumping knowledge networks”. In: *International conference on machine learning*. PMLR, pp. 5453–5462.
- Yang, Bishan et al. (2014). “Embedding entities and relations for learning and inference in knowledge bases”. In: *CoRR*.

- Zhang, Shuai et al. (2019). “Quaternion knowledge graph embeddings”. In: *Advances in neural information processing systems* 32.
- Zhang, Siheng, Zhengya Sun, and Wensheng Zhang (2020). “Improve the translational distance models for knowledge graph embedding”. In: *Journal of Intelligent Information Systems* 55, pp. 445–467.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won’t result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
 - (b) Complete proofs of all theoretical results. [Not Applicable]
 - (c) Clear explanations of any assumptions. [Not Applicable]
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Not Applicable]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. [Not Applicable]
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

7 SQUARE LORENTZ DISTANCE VS. GEODESIC DISTANCE

In this section, we illustrate that the score function based on the utilized Lorentz distance behaves similarly to the geodesic distance for different t . In particular, in Figure 8, we present the two-dimensional landscapes of the TPTF score function for $t \in \{-10, 0, 10\}$.

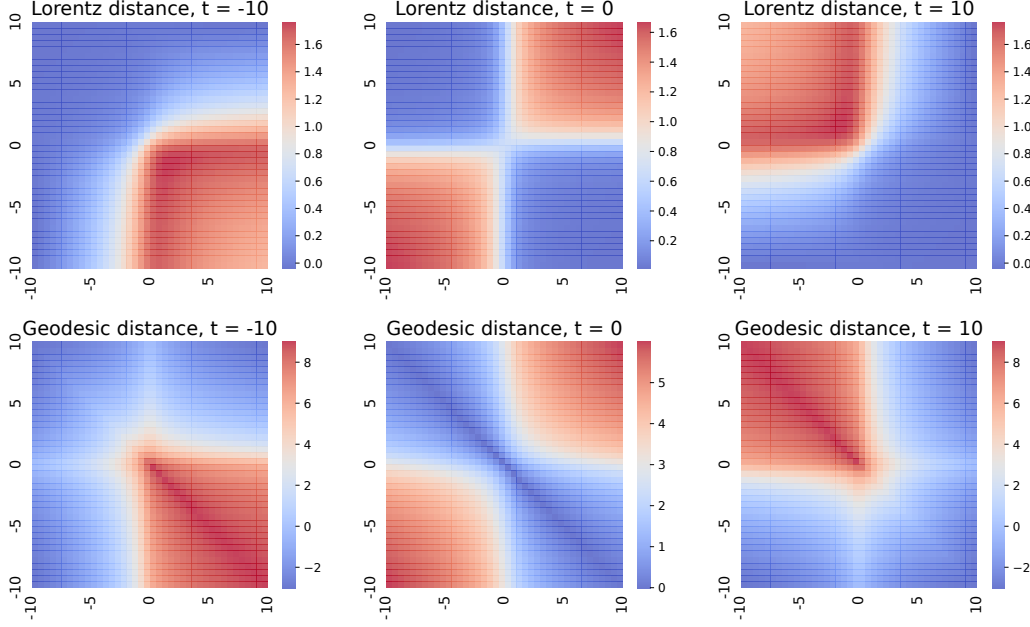


Figure 8: The landscapes of the TPTF score function for the Lorentz and geodesic distances. As is seen in the plots, the normalized score function with the Lorentz distance mimics the behaviour of ones with geodesic distance.

8 COMPLEXITY

In this section, we estimate the complexity of our MIG-TF model.

Let us introduce the operation $G \times_i V$ for a tensor $G \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_n}$ and matrix $V \in \mathbb{R}^{d_i \times m}$

$$(G \times_i V)_{a_1, \dots, a_{i-1}, k, a_{i+1}, \dots, a_n} = \sum_{a_i=1}^{d_i} G_{a_1, a_2, \dots, a_n} V_{a_i, k}.$$

Let n_e and n_r be respectively the numbers of entities and relations in a knowledge graph, b – a batch size, d_e and d_r – embedding sizes of the Tucker model, d_h – the size of hyperbolic embeddings. Let the core of the Tucker decomposition be $G \in \mathbb{R}^{d_e \times d_e \times d_r}$, the matrices of embeddings – the entities embedding matrix $U \in \mathbb{R}^{b \times d_e}$, the relation embedding matrix $T \in \mathbb{R}^{b \times d_r}$ and another entity embedding matrix $V \in \mathbb{R}^{n_e \times d_e}$. Similarly, the matrices in the hyperbolic model are $U^{\mathcal{L}} \in \mathbb{R}^{b \times d_h}$, $T^{\mathcal{L}} \in \mathbb{R}^{b \times d_h}$ and $V^{\mathcal{L}} \in \mathbb{R}^{n_e \times d_h}$.

Firstly, we find the complexity of the Tucker model. To compute the Tucker score function, we firstly find $G_1 = G \times_1 U \in \mathbb{R}^{b \times d_e \times d_r}$. It takes $O(bd_e(d_e d_r))$ operations. Then we compute $G_2 = G_1 \times_3 T \in \mathbb{R}^{b \times d_e \times b}$ with $O(bd_r(bd_e))$ operations. Finally, the diagonal slice of G_2 is taken and multiplied with V with $O(bd_e n_e)$ operations. Overall, the total number of operations is:

$$O_{Tucker} = O(bd_e(d_e d_r) + bd_r(bd_e) + bd_e n_e) = O(bd_e(d_e d_r + bd_e + n_e)).$$

Secondly, we find the complexity of the TPTF model. First of all, we compute Lorentz zero element of embeddings x as

$$x_0 = \sqrt{\beta + \sum_{i=1}^d x_i^2},$$

which is $O(d_h(b + n_e))$ operations. Then we compute Lorentz inner products and it takes $O(bd_hn_e + bd_hn_e + bd_hn_e) = O(bd_hn_e)$. Then we compute the sum of all inner products S_H using $O(bn_e)$ operations. Finally, we compute the normalization N_H with $O(bn_e)$ operations and make elementwise division S_H/N_H with the complexity $O(bn_e)$. Overall, the TPTF complexity is:

$$O_{TPTF} = O(d_h(b + n_e) + bd_hn_e + bn_e + bn_e + bn_e) = O(bn_ed_h).$$

The complexity of MIG-TF iteration is a sum of TuckER and TPTF complexities and is equal to:

$$O_{MIG-TF} = O_{TuckER} + O_{TPTF} + O(bn_e) = O(bd_e(d_e d_r + bd_e + n_e) + bn_e d_h).$$

In terms of time and parameter complexity, our MIG-TF model adds only 30% overhead on training time and 25% overhead on inference time in comparison to TuckER.

9 MIXED GEOMETRY TENSOR FACTORIZATION MODIFICATIONS

In this section, we describe in more detail additional techniques that we use for our MIG-TF model.

9.1 Riemannian MIG-TF

To improve the quality of the TuckER (Hospedales, 2019) model we use the so-called *RTuckER* model (Peshekhonov, Arzhantsev, and Rakhuba, 2024). This model has the same architecture as TuckER, but was trained with the methods of Riemannian optimisation. This Riemannian model significantly outperforms TuckER on some knowledge graphs. For more details we refer to (Peshekhonov, Arzhantsev, and Rakhuba, 2024). We utilize *RTuckER* as a Euclidean term of our MIG-TF model only on WN18RR knowledge graph on which the Riemannian model significantly outperforms TuckER.

9.2 MIG-TF with QR decomposition

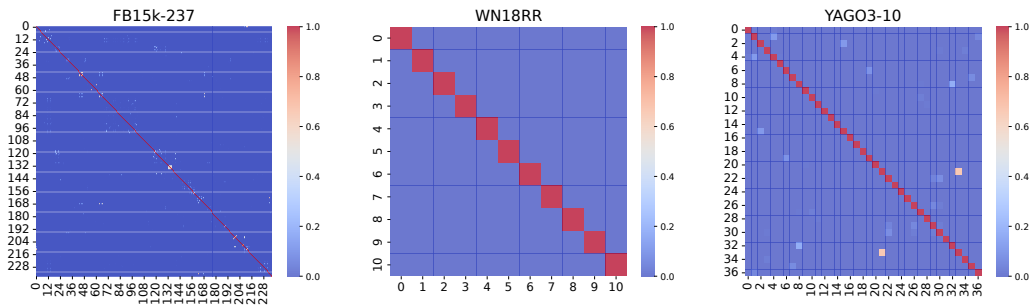


Figure 9: Correlation maps for relations of knowledge graphs FB15k-237, WN18RR and YAGO3-10. As can be seen the vast majority of relations in knowledge graphs are uncorrelated. Therefore, the orthogonality of relation embeddings is promising as embeddings of uncorrelated relations do not correlate.

As was shown in Figure 9, most of the relations in knowledge graphs have no correlations, therefore maintaining the orthogonality of relation embeddings creates better vector representation. For this purpose, on each iteration we update the hyperbolic relation embedding matrix $T^{\mathcal{L}}$ with the help of the QR decomposition as follows:

$$T^{\mathcal{L}} = Q^{\mathcal{L}} R. \quad (8)$$

Then we compute the score function using $Q^{\mathcal{L}}$ instead of $T^{\mathcal{L}}$:

$$(S_{MIG-TF})_i = S_E(G, u, V_i, t) + S_H(u^\mathcal{L}, V_i^\mathcal{L}, q^\mathcal{L}).$$

In these formulas, G is the core of Tucker decomposition, u and $u^\mathcal{L}$ are the Euclidean and hyperbolic entity embeddings, V_i and $V_i^\mathcal{L}$ are respectively the i -th rows of the Euclidean V and of the hyperbolic $V^\mathcal{L}$ entity embedding matrices. The vector t is the Euclidean relation embedding and $q^\mathcal{L}$ is a row of $Q^\mathcal{L}$ from (8). We call this modified MIG-TF as $MIG-TF_{QR}$.

The MIG-TF model with QR decomposition significantly outperform MIG-TF on knowledge graphs WN18RR and YAGO3-10 where the condition of non correlation of relations is fulfilled almost everywhere. However, $MIG-TF_{QR}$ slightly loses to MIG-TF on FB15k-237 graph, where the condition of non correlation is usually violated (see Figure 9).

10 ROBUSTNESS

To analyze our models' robustness, we add random links in the training data knowledge graph. If n_{train} is the number of links in train data, then we add αn random links to train data. We train the mixed geometry model on poisoned training data and test it on the test data, see Figure 10. As we see, even with high levels of noise (10%), both MIG-TF and $MIG-TF_{QR}$ models do not significantly degrade.

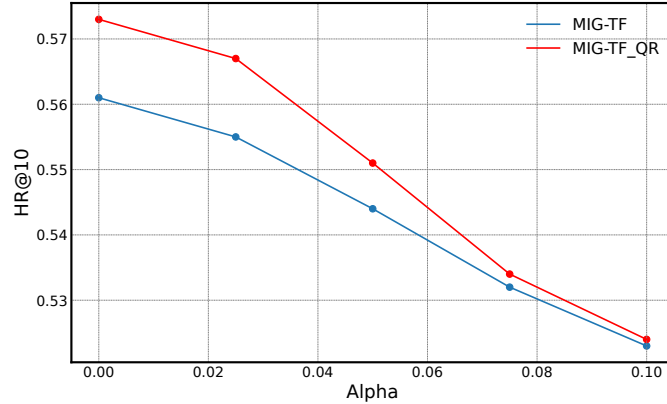


Figure 10: The HR@10 metric on WN18RR knowledge graph depended on the fraction of poison α in training data for 2 mixed geometry models.

11 THE IMPACT OF EUCLIDEAN AND HYPERBOLIC TERMS

In this section, we analyze impact of different terms in the MIG-TF score function. We represent the score function S_{MIG-TF} as:

$$S_{MIG-TF} = \mu S_E + (1 - \mu) S_H, \quad \mu \in (0, 1),$$

where S_E and S_H are respectively Euclidean and hyperbolic components of the score function and μ is a parameter. By changing μ , we measure the effect of the score function components on the HR@10 metric on WN18RR knowledge graph, see Figure 11. It is interesting to note that the best quality is achieved with the equal effect of the Euclidean and hyperbolic terms. This justifies our choice of the score function.

12 OPTIMAL CURVATURE

This section demonstrates dependence of link prediction quality on WN18RR knowledge graph on the curvature in hyperbolic parts of TPTF and MIG-TF models, see Figure 12. Note that the optimal curvature for hyperbolic and mixed geometry models are different from each other.

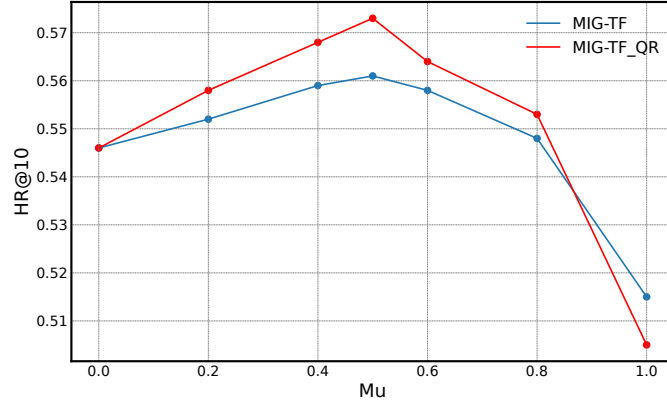


Figure 11: HR@10 against the μ parameter for 2 mixed geometry models. The maximum of HR@10 metrics on WN18RR is achieved in $\mu = 0.5$, where the contributions of Euclidean and hyperbolic terms of score function are equal.

13 HYPERPARAMETERS

In this section, we present the hyperparameters of both the Euclidean term (Table 4) and the hyperbolic term (Table 6) of our MIG-TF model. In this tables, d_e, d_r are the sizes of entity and relation embeddings, lr is a learning rate, $\mathcal{N}(0, \rho_e^2 I), \mathcal{N}(0, \rho_r^2 I)$ – distributions of initial weights of entities and relations embeddings in the TPTF component of the hybrid model and β is a curvature of hyperboloid in the Lorentz term of MIG-TF model. All the experiments we performed were done on one GPU NVIDIA V100.

Table 4: Hyperparameters for the Tucker component.

Hyperparametr	WN18RR	FB15k-237	YAGO3-10
d_e	200	200	200
d_r	30	200	30
lr	0.01	0.001	0.003
$epochs$	500	500	500
$Dropout_1$	0.2	0.3	0.2
$Dropout_2$	0.2	0.4	0.2
$Dropout_3$	0.3	0.5	0.2

Table 5: Hyperparameters for the TPTF component.

Hyperparametr	WN18RR	FB15k-237	YAGO3-10
$d_{\mathcal{L}}$	50	50	50
lr	0.003	0.002	0.005
β	1.3	1	1.1
$epochs$	250	150	250
ρ_e	0.005	0.001	0.001
ρ_r	0.005	0.001	0.001
$Dropout_e$	0.2	0.3	0.2
$Dropout_r$	0.2	0.3	0.2

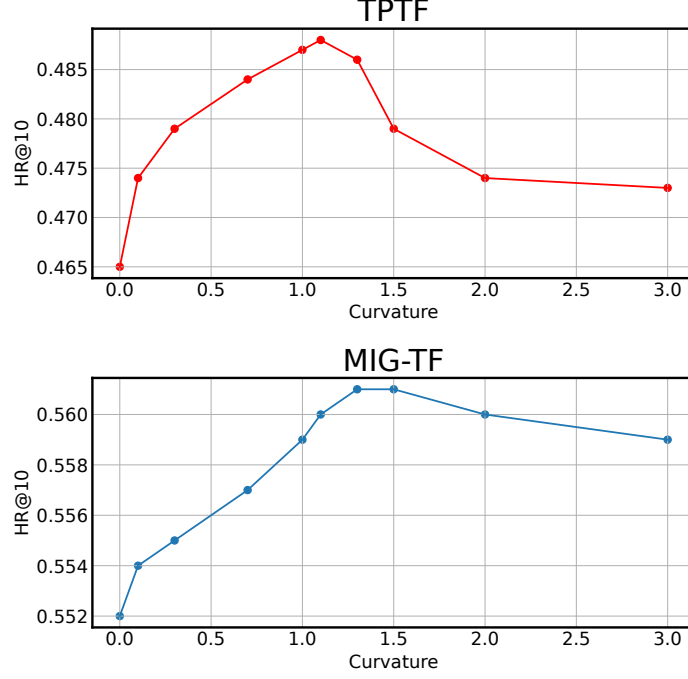


Figure 12: The top graph presents the dependence of $HR@10$ metric on WN18RR knowledge graph from curvature of Hyperbolic space β for TPTF model. The bottom one presents the dependence of metric from curvature for MIG-TF model.

Table 6: Hyperparameters for the MIG-TF component.

Hyperparametr	WN18RR	FB15k-237	YAGO3-10
$d_{\mathcal{L}}$	50	50	50
lr	0.003	0.001	0.005
β	1.5	1	1.1
$epochs$	250	100	150
ρ_e	0.005	0.001	0.001
ρ_r	0.005	0.001	0.001
$Dropout_e$	0.2	0.3	0.2
$Dropout_r$	0.2	0.3	0.2