Low-Dimensional Knowledge Graph Embeddings via Hyperbolic Rotations

Ines Chami, Adva Wolf, Frederic Sala & Christopher Ré
Department of Computer Science
Stanford University
{chami,advaw,fredsala}@stanford.edu, chrismre@cs.stanford.edu

Abstract

Knowledge graphs (KGs) capture rich relationships between a large number of entities. Embeddings of these structures must preserve these relationships with high fidelity. Recently, hyperbolic embedding methods have achieved state-of-the-art quality in graph representation learning tasks; when embedding certain graphs, they can produce parsimonious embeddings that have higher fidelity while using much fewer dimensions than their Euclidean counterparts. Mirroring work in Euclidean space, we are the first to leverage trainable hyperbolic rotations, a key notion in providing sufficiently rich representations for the complex logical patterns found in KGs. Coupled with trainable curvature, this approach yields improved embeddings in fewer dimensions. We evaluate our method, ROTATIONH, on the WN18RR link prediction task; in the low-dimensional setting, we improve on previous Euclidean-based efforts by 2.2% in mean reciprocal rank (MRR), and in the high-dimensional setting, we achieve a new state-of-the-art MRR of 49.2%, improving on existing methods by 1.1%.

1 Introduction

Knowledge graphs are popular data structures for representing factual knowledge to be queried and used in downstream applications such as word sense disambiguation, question answering, and information extraction. A common approach to encoding the entities and relationships in KGs is via embeddings into vector spaces [12, 4, 19]; these methods have achieved promising results for many tasks. However, preserving the complex relationships while limiting the memory use (or equivalently, the dimensionality) of the embeddings, is an important challenge.

At first glance, these desiderata appear to be in tension with each other. The expressiveness of embeddings is typically improved by increasing their dimensionality, thus expanding the memory required to store the KG. For hierarchical data, exciting recent works have proposed embedding into hyperbolic space instead of conventional Euclidean space [13, 15]—which alleviates this tension. Hyperbolic space embeds trees with arbitrarily low distortion in just two dimensions; this is because hyperbolic geometry resembles a continuous version of trees. However, such works focus on embedding simpler graphs (e.g., weighted trees) and cannot express the diverse and complex relationships in KGs.

We propose a new approach that seeks to achieve the best of both worlds——the parsimonious representations offered by hyperbolic space, along with the rich transformations needed to encode and infer logical patterns in KGs, previously only defined in Euclidean space. To do so, we parametrize rotations and translations of hyperbolic space, enabling us to learn relationship-specific hyperbolic transformations. In addition, because the optimization of hyperbolic models is known to be challenging, we apply two recent techniques [5] to minimize this pain point: trainable curvature and tangent space optimization. We evaluate the performance of ROTATIONH on the KG completion task using the standard WN18RR [6, 3] dataset (Section 5). In low (32) dimensions, we improve over

Graph Representation Learning Workshop (NeurIPS 2019), Vancouver, Canada.

Euclidean-based models by 2.2% in the MRR metric. Additionally, at high (500) dimensions, we achieve new state-of-the-art results, obtaining 49.2% MRR, improving previous methods by 1.1%.

2 Related Work

Euclidean embeddings. In the past decade, there has been a rich literature on Euclidean embeddings for KG representation learning. These include translation approaches [3, 8, 20, 11] or tensor factorization methods [12, 21]. While these methods are fairly simple and have few parameters, they fail to encode important logical properties (e.g., translations can't encode symmetry).

Euclidean complex embeddings. More recently, there has been interest in learning embeddings in complex space, as in the ComplEx [18] and RotatE [16] models. RotatE learns rotations in $\mathbb{C}^d$, which are very effective in capturing multiple logical properties. However, a downside is that these embeddings require high-dimensional spaces, leading to high memory costs.

Hyperbolic embeddings. To the best of our knowledge, MuRP [1] is the only method learning KG embeddings in hyperbolic space, targeting hierarchical data. MuRP minimizes hyperbolic distances between a re-scaled version of the head entity embedding and a translation of the tail entity embedding. It achieves promising results using hyperbolic embeddings with fewer dimensions than its Euclidean analogues. However, MuRP is a translation model and fails to encode some logical properties of relationships. Furthermore, embeddings are learned in a hyperbolic space with fixed curvature, leading to insufficient precision, and it relies on cumbersome Riemannian optimization.

3 Background and Problem Setting

Problem Setting. We have access to a training set of triples $(h, r, t) \in \mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, where $\mathcal{V}$ and $\mathcal{R}$ are entity and relationship sets, respectively. The goal in KG embeddings is to map entities $(h, t) \in \mathcal{V} \times \mathcal{V}$ to embeddings $(\mathbf{e}_h, \mathbf{e}_t) \in \mathcal{U}^{n_\mathcal{V}} \times \mathcal{U}^{n_\mathcal{V}}$, and relationships $r \in \mathcal{R}$ to embeddings $\mathbf{r}_r \in \mathcal{U}^{n_\mathcal{R}}$, for some choice of space $\mathcal{U}$ (traditionally $\mathbb{R}$), such that the KG structure is preserved. In the link prediction task, the data is split into $\mathcal{E}_{Train}$ and $\mathcal{E}_{Test}$ triples. Embeddings are usually learned by optimizing a scoring function $s : \mathcal{V} \times \mathcal{R} \times \mathcal{V} \to \mathbb{R}$, which measures triples' likelihoods. $s(\cdot, \cdot, \cdot)$ is trained using triples in $\mathcal{E}_{Train}$ and the learned embeddings are then used to predict scores for triples in $\mathcal{E}_{Test}$. The goal is to learn embeddings such that the scores of triples in $\mathcal{E}_{Test}$ are high compared to triples that are not present in $\mathcal{E}$.

Review of Hyperbolic Geometry. We briefly review key notions from hyperbolic geometry; a more in-depth treatment is available in standard texts [14]. Our hyperbolic model is the n-dimensional Poincaré ball with curvature $-c$ ($c > 0$): $\mathbb{B}^{n,c} = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}||^2 < \frac{1}{c}\}$. For each point $\mathbf{x} \in \mathbb{B}^{n,c}$, we map the tangent space $\mathcal{T}^c_\mathbf{x}$, which provides a first-order approximation of the manifold $\mathbb{B}^{n,c}$ at $\mathbf{x}$, to $\mathbb{B}^{n,c}$ via the exponential map, $\exp^c_\mathbf{x}(\mathbf{v}) = \mathbf{x} \oplus^c \left(\tanh\left(\frac{\sqrt{c}||\mathbf{v}||}{1-c||\mathbf{x}||^2}\right)\frac{\mathbf{v}}{\sqrt{c}||\mathbf{v}||}\right)$, where

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

denotes the Mobïus addition [7]. In contrast to Euclidean addition, it is not commutative, but provides an analogue through the lens of parallel transport: given two points $\mathbf{x}, \mathbf{y}$ and a vector $\mathbf{v}$ in $\mathcal{T}^c_\mathbf{x}$, there is a unique vector in $\mathcal{T}^c_\mathbf{y}$ which creates the same angle as $\mathbf{v}$ with the direction of the geodesic connecting $\mathbf{x}$ to $\mathbf{y}$. This map is the parallel transport $P^c_{\mathbf{x}\to\mathbf{y}}(\cdot)$; Euclidean parallel transport is the standard Euclidean addition. Analogously, Mobïus addition satisfies [7] $\mathbf{x} \oplus^c \exp^c_\mathbf{0}(\mathbf{v}) = \exp^c_\mathbf{x}(P^c_{\mathbf{0}\to\mathbf{x}}(\mathbf{v}))$. Finally, the hyperbolic distance on $\mathbb{B}^{n,c}$ has the explicit formula $d^c(\mathbf{x}, \mathbf{y}) = 2\text{artanh}(\sqrt{c}|| - \mathbf{x} \oplus^c \mathbf{y}||)/\sqrt{c}$.

4 KG Embeddings via Hyperbolic Rotations

Our goal is to learn hyperbolic embeddings that can encode complex logical relationships. We introduce ROTATIONH, a hyperbolic embedding model for multi-relational data. ROTATIONH learns embeddings in hyperbolic space and models relation-specific transformations with hyperbolic rotations (Section 4.1), followed by translations. All ROTATIONH parameters are defined in the Euclidean tangent space at the origin and can be learned with standard Euclidean optimization techniques (Section 4.2), which are empirically more stable than Riemannian optimization methods.

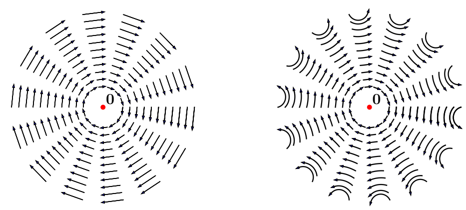

Figure 1: Euclidean (left) and hyperbolic (right) rotations. As two points move away from the origin in hyperbolic space, their distance approaches the sum of distances between the points and the origin. This resembles trees, where the shortest path goes through the nearest common ancestor.

4.1 Rotations via Givens transformations

Relationships often satisfy particular properties, such as symmetry: i.e., if (*Michelle Obama*, *married to*, *Barack Obama*) holds, then (*Barack Obama*, *married to*, *Michelle Obama*) does as well. These rules are not universal. For instance, (*Barack Obama*, *born in*, *Hawaii*) is not symmetric. Creating and curating a set of deterministic rules is infeasible; instead we parametrize and learn such properties. A convenient vehicle to do so is *rotations*, which have been successfully used to encode logical properties in complex space [16]. We learn relationship-specific rotations using Givens transformations: 2×2 rotation matrices commonly used in numerical linear algebra. We assume an even number of dimensions and parametrize rotations with block-diagonal rotation matrices of the form:

$$R(\Theta_r) = \text{diag}(G(\theta_{r,1}), G(\theta_{r,2}), \ldots, G(\theta_{r,\frac{n}{2}})), \text{ where } G(\theta) := \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad (2)$$

and $\Theta_r := (\theta_{r,i})_{i \in \{1,\ldots \frac{n}{2}\}}$ are relationship-specific parameters. Note that rotations of this form are centered at zero and are isometries of the hyperbolic space (distance-preserving). We can therefore directly apply this transformation to hyperbolic embeddings while preserving the underlying geometry. We illustrate two-dimensional rotations in both Euclidean and hyperbolic spaces in Figure 1.

4.2 Optimization

We let $(\mathbf{e}_v^H)_{v \in \mathcal{V}}$ and $(\mathbf{r}_r^H)_{r \in \mathcal{R}}$ denote entity and relationship hyperbolic embeddings respectively. MuRP learns embeddings via Riemannian Stochastic Gradient Descent (RSGD) [2]. In practice, optimization in hyperbolic space is challenging, requiring the use of sophisticated techniques and special care not to fall off the manifold. We overcome this obstacle by defining all the ROTATIONH parameters in the Euclidean tangent space at the origin, thanks to the exponential map. In other words, ROTATIONH optimizes Euclidean entity and relationship embeddings $(\mathbf{e}_v^E)_{v \in \mathcal{V}}$ and $(\mathbf{r}_r^E)_{r \in \mathcal{R}}$, which are mapped to the Poincaré ball with $\mathbf{e}_v^H = \exp_{\mathbf{0}}^c(\mathbf{e}_v^E)$ and $\mathbf{r}_r^H = \exp_{\mathbf{0}}^c(\mathbf{r}_r^E)$.

4.3 ROTATIONH model

We have all of the building blocks for ROTATIONH, and can now provide the model architecture. For a triple $(h, r, t) \in \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, ROTATIONH first maps Euclidean embeddings to hyperbolic embeddings, and applies a relation-specific rotation (Equation 2) to the head entity embedding, followed by a hyperbolic translation (Equation 1). The resulting scoring function is:

$$s(h, r, t) = -d_c((R(\Theta_r)\exp_{\mathbf{0}}^c(\mathbf{e}_h^E)) \oplus_c \exp_{\mathbf{0}}^c(\mathbf{r}_r^E), \exp_{\mathbf{0}}^c(\mathbf{e}_t^E))^2 + b_h + b_t, \quad (3)$$

where $(b_v)_{v \in \mathcal{V}}$ are entity biases which act as a margin in the scoring function [17, 1]. The model parameters are then $\{(\Theta_r)_{r \in \mathcal{R}}, (\mathbf{r}_r^E)_{r \in \mathcal{R}}, (\mathbf{e}_v^E)_{v \in \mathcal{V}}, (b_v)_{v \in \mathcal{V}}\}$, which are all Euclidean parameters and can be learned using standard Euclidean optimization techniques. Note that we also learn the absolute hyperbolic curvature with $c = \text{Softplus}(c') \in \mathbb{R}_+$; as we shall see, this is critical. We train the ROTATIONH model by minimizing the cross-entropy loss over true triples and negative triples using negative sampling: for each positive example, we sample S negative examples by randomly corrupting the head and the tail entities:

$$\mathcal{L} = \frac{1}{|\mathcal{E}_{Train}|(1+S)} \sum_{(h,r,t) \in \mathcal{E}_{Train}} \left[\log\left(1 + \mathrm{e}^{-s(h,r,t)}\right) + \sum_{\substack{k=1 \\ h'_k, t'_k \sim \text{unif}(\mathcal{V})}}^{S} \log\left(1 + \mathrm{e}^{s(h'_k, r, t'_k)}\right) \right]. \quad (4)$$

Space $\mathcal{U}$	Model	MRR	H@1	H@3	H@10
$\mathbb{R}^n$	MuRE [1]	.458	.421	.471	.525
$\mathbb{C}^n$	ComplEx-N3 [10]	.420	.390	.420	.460
$\mathbb{B}^{n,1}$	MuRP [1]	.465	.420	.484	.544
$\mathbb{R}^n$	ROTATIONE	.448	.408	.463	.522
$\mathbb{B}^{n,c}$	ROTATIONH	**.470**	**.428**	**.487**	**.547**
$\mathbb{R}^n$	TransE [3]	.226	-	-	.501
	DistMult [21]	.430	.390	.440	.490
	ConvE [6]	.430	.400	.440	.520
	MuRE [1]	.475	.436	.487	.554
$\mathbb{C}^n$	ComplEx [18]	.440	.410	.460	.510
	ComplEx-N3 [10]	.480	-	-	.570
	RotatE [16]	.476	.428	.492	.571
$\mathbb{B}^{n,1}$	MuRP [1]	.481	.440	.495	.566
$\mathbb{R}^n$	ROTATIONE	**.494**	**.446**	**.512**	.585
$\mathbb{B}^{n,c}$	ROTATIONH	.492	.443	.510	**.586**

Table 1: Link prediction results for WN18RR, for low dimensions ($n = 32$, top five rows), and high dimensions ($n \geq 200$, bottom rows). Best in **bold** and best published underlined.

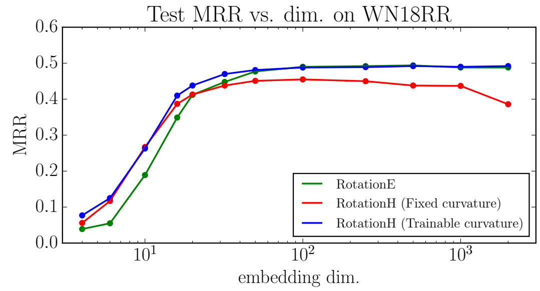

Figure 2: ROTATIONH offers improved performance at low dimensions; at high dimensions, fixed curvature degrades performance, while trainable curvature recovers the Euclidean space.

5 Link Prediction Experiments

We hypothesize that when the data has a hierarchical structure, ROTATIONH obtains better representations and allows for improved downstream performance at fewer dimensions than comparably expressive Euclidean models. We also expect ROTATIONH to improve on translation-based hyperbolic models like MuRP.

Dataset. We evaluate ROTATIONH on the link prediction task on the WN18RR dataset, often used as a competition benchmark. WN18RR is a subset of WordNet containing relationships between words, and has a natural hierarchical structure, e.g., (*car*, *hypernym of*, *sedan*). WN18RR consists of 40,943 entities and 11 relations, with a total of 86,835 train, 3034 validation and 3,134 test triples.

Protocol. We learn hyperbolic embeddings with ROTATIONH by minimizing the loss function in Equation 4. At test time, we use the scoring function in Equation 3 to rank the correct tail or head entity against all possible entities and compute two ranking-based metrics: (1) MRR, which measures the mean of inverse ranks assigned to correct entities, and (2) Hits at K (H@K, $K \in \{1, 3, 10\}$), which measures the proportion of correct triples among top K predicted triples. We follow the standard evaluation protocol in the filtered setting [3]: all true triples in the KG are filtered out during evaluation, since predicting a low rank for these triples shouldn't be penalized. We implement our model in PyTorch and will make our code publicly available. We train with Adam [9], sweeping hyper-parameters using the validation set over learning rate (α), dimension (n), negative sample size (S) and batch size (B). Our best model uses $\alpha = 10^{-3}, n = 500, S = 50$ and $B = 500$.

Results. We report link prediction results on WN18RR in Table 1. We compare ROTATIONH to state-of-the-art methods as well as its Euclidean analogue (ROTATIONE, which is ROTATIONH for $c = 0$). In low dimensions (top rows), ROTATIONH improves over ROTATIONE by 2.2% points in MRR. In the high-dimensional regime (lower rows), we compare against a variety of other models, including the recent hyperbolic (but not rotation-based) MuRP, observing that ROTATIONH achieves new state-of-the-art results.

Dimension and Curvature. We further analyze the role of the dimension by plotting MRR versus the embedding dimension in Figure 2. As expected, the benefit of hyperbolic embeddings (over Euclidean embeddings) is prominent in the low-dimensional regime ($n < 100$), where ROTATIONH with fixed and trainable curvature both offer better performance than ROTATIONE.

In the high-dimensional regime ($n > 100$), we note that both ROTATIONE and ROTATIONH with trainable curvature have similar performance, while ROTATIONH with fixed curvature does worse—confirming the importance of trainable curvature and its impact on precision and capacity, formerly studied in [15]. Empirically, we observe that the absolute curvature is small in high dimensions while it is closer to one in low dimensions. We conjecture that when the dimension is sufficiently large, Euclidean space has enough capacity to encode complex interactions between entities. ROTATIONH performs as well or better than previous methods in both low- and high-dimensional regimes.

Acknowledgments

We gratefully acknowledge the support of DARPA under Nos. FA87501720095 (D3M), FA86501827865 (SDH), and FA86501827882 (ASED); NIH under No. U54EB020405 (Mobilize), NSF under Nos. CCF1763315 (Beyond Sparsity), CCF1563078 (Volume to Velocity), and 1937301 (RTML); ONR under No. N000141712266 (Unifying Weak Supervision); the Moore Foundation, NXP, Xilinx, LETI-CEA, Intel, Microsoft, NEC, Toshiba, TSMC, ARM, Hitachi, BASF, Accenture, Ericsson, Qualcomm, Analog Devices, the Okawa Foundation, American Family Insurance, Google Cloud, Swiss Re, TOTAL, and members of the Stanford DAWN project: Teradata, Facebook, Google, Ant Financial, NEC, VMWare, and Infosys. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views, policies, or endorsements, either expressed or implied, of DARPA, NIH, ONR, or the U.S. Government.

References

[1] Ivana Balažević, Carl Allen, and Timothy Hospedales. Multi-relational Poincaré graph embeddings. In *Advances in Neural Information Processing Systems*, 2019.

[2] Silvere Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[4] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 301–306, 2011.

[5] Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2019.

[6] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, 2018.

[8] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun 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)*, pages 687–696, 2015.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015.

[10] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. *International Conference on Machine Learning*, 2018.

[11] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI Conference on Artificial Intelligence*, 2015.

[12] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, pages 809–816. Omnipress, 2011.

[13] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6338–6347, 2017.

[14] Joel W Robbin and Dietmar A Salamon. Introduction to differential geometry.

[15] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, pages 4457–4466, 2018.

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

[17] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré GloVe: Hyperbolic word embeddings. In *International Conference on Learning Representations*, 2019.

[18] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080, 2016.

[19] Q Wang, Z Mao, B Wang, and L Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743, December 2017.

[20] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

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