Designing Hierarchies for Optimal Hyperbolic Embedding

Melika Ayoughi[0000−0002−6847−5999], Max van Spengler[0000−0002−7440−920X], Pascal Mettes[0000−0001−9275−5942], and Paul Groth[0000−0003−0183−6910]

University of Amsterdam

Abstract. Hyperbolic geometry has shown to be highly effective for embedding hierarchical data structures. As such, machine learning in hyperbolic space is rapidly gaining traction across a wide range of disciplines, from recommender systems and graph networks to biological systems and computer vision. The performance of hyperbolic learning commonly depends on the hierarchical information used as input or supervision. Given that knowledge graphs and ontologies are common sources of such hierarchies, this paper aims to guide ontology designers in designing hierarchies for use in these learning algorithms. Using widely employed measures of embedding quality with extensive experiments, we find that hierarchies are best suited for hyperbolic embeddings when they are wide, and single inheritance, independent of the hierarchy size and imbalance.

Keywords: Hyperbolic Learning · Ontology Design · Machine Learning

1 Introduction

Knowledge graphs and ontologies provide a rich source of hierarchical information, such as the classification of creative works in Schema.org [1] or the organization of professions in Wikidata [2]. This hierarchical structure is well-suited for machine learning, particularly in hyperbolic learning, which utilizes hyperbolic geometry to embed tree-like structures into low-dimensional spaces [45]. Such embeddings have been shown to enhance performance in tasks like image and video classification [29,36,37], audio understanding [24,52], and recommender systems [34,62,64]. Throughout the literature [45,49], hyperbolic representation learning approaches typically assume that hierarchies are provided *as is*. However, ontology engineers consider multiple factors when designing ontologies beyond representing the domain (e.g. reusing existing ontologies, use concepts for interoperability, end user tasks). This work seeks to offer insights for ontology engineers in crafting hierarchies optimized for hyperbolic hierarchical embeddings. Unlike previous studies, which focus on improving hyperbolic embeddings for a given hierarchy, we address the reverse question: how can hierarchies be designed to enhance their suitability for embedding in hyperbolic space? Specifically, we conduct controlled experiments to examine how different tree structures affect

[1] https://schema.org/

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page

the quality of embeddings produced by two primary classes of hyperbolic embedding algorithms: gradient-based and construction-based methods. Embedding quality is evaluated using distortion metrics [55], which quantify the discrepancy between distances in the embedding space (calculated via a continuous distance function) and the original graph distances (defined by the edge count between nodes). Our objective is to uncover the key axes of hierarchy design that influence the effectiveness of hyperbolic embeddings. Our results demonstrate that hierarchies optimized for width, rather than height, are best suited for hyperbolic embeddings. Hierarchy imbalance and size are shown to have minimal impact, while multiple inheritance should be avoided. We validate these findings using a real-world scenario, where alternative semantic organizations significantly reduce distortion. These results complement existing approaches to ontology design and evaluation by providing actionable insights for ontology engineers to enhance downstream embedding quality. We hope these recommendations assist ontology designers when downstream hyperbolic embedding performance is a priority. In summary, the contributions of this paper are as follows:

- **In-depth empirical study:** We perform in-depth analyses across four hyperbolic embedding algorithms to examine the impact of hierarchy structure on embedding quality.
- **Practical recommendations:** Our experiments lead to four recommendations for ontology engineers on structuring hierarchical portions of ontologies.
- **Real-world case study:** We validate our recommendations on real-world use cases, highlighting the inherent trade-off between ontological design goals and downstream utility in continuous hyperbolic spaces.

Our recommendations apply to various real-world scenarios requiring hierarchical data in continuous spaces, such as: recommender systems (e.g. product/content hierarchies), drug discovery (e.g., gene Ontology, SNOMED CT), and biological analysis (e.g. protein families). Code is openly available here [3].

2 Related Work

2.1 Hierarchy and ontology design

Hierarchies, particularly taxonomic backbones, whether formal or informal, play a critical role in ontology and knowledge graph design [22,28,44]. They organize complex domains [57] into manageable components and enable various forms of reasoning, such as subsumption. Changes in hierarchy structure can have a significant downstream impact on applications [51]. Therefore, ontology design methodologies provide guidance on crafting hierarchies to reflect domain constraints and ensure proper reasoning outcomes [28,53].

Works focused on the evaluation of ontologies also consider hierarchy [20,43]. These studies assess aspects such as whether a hierarchy correctly partitions instances, whether there are cycles of specialization and generalization, and

[3] https://github.com/Melika-Ayoughi/Optimal-Hierarchy

whether instance assertions are semantically accurate [20]. Other evaluation approaches adopt principled criteria and metrics based on formal notions (e.g. unity) [3,17,18]. Examples of criteria include the complexity of the hierarchy (e.g. number of classes, depth, number of top level classes) as well as conciseness (e.g. cycles, and classes without instances) [50]

In ontology induction and knowledge graph construction, the creation of high-quality hierarchies is a critical consideration [69,63]. Evaluation typically involves expert review, comparison with gold-standard ontologies, or the application of aforementioned established evaluation criteria. Our work complements these existing recommendations, metrics, and evaluation approaches [43] by providing ontology engineers with guidance on designing hierarchical structures to enable machine learning tasks.

2.2 Learning over knowledge graphs using hierarchies

A significant body of work leverages hierarchical information within knowledge graphs to enhance machine learning tasks [27]. These tasks include link prediction [6,70], question answering [11], and query answering using embedding spaces [25]. Additionally, research has focused on creating embeddings for knowledge graphs with complex semantics [5]. Our work differs by providing guidance to ontology engineers on designing hierarchies, rather than focusing on embedding design for existing knowledge graphs.

2.3 Hyperbolic representation learning

We focus on hyperbolic embeddings, because they demonstrate superior performance in representing hierarchical data structures compared to Euclidean methods. In early work, Sarkar [56] introduced Delaunay tree embeddings in hyperbolic space, demonstrating the potential of hyperbolic geometry to achieve tree embeddings with arbitrarily low distortion. However, Sarkar's construction-based algorithm is limited to 2D embeddings, reducing its expressiveness and applicability in deep learning contexts. To address this limitation, Nickel and Kiela [46] proposed a contrastive approach that supports embedding optimization in any dimensionality, significantly outperforming Euclidean embeddings on trees. This line of work has been extended through entailment cones to induce partial hierarchical order [14,66], adapted to the Lorentz model of hyperbolic space [32,47], and improved by incorporating distortion [67] or separation [38] objectives during optimization. Subsequently, Sala et al. [55] expanded Sarkar's construction-based approach to higher-dimensional embeddings. Overall, these algorithms, whether optimized via gradient descent or constructed explicitly, consistently outperform Euclidean embeddings. This superiority stems from the insight that "hyperbolic space can be thought of as a continuous analogue to discrete trees" [47], owing to their shared nature of exponential growth.

In light of the strong performance of hyperbolic representation learning, numerous studies have integrated hyperbolic embeddings into neural networks, enabling deep learning to incorporate hierarchical knowledge. Hyperbolic learning have been shown to improve recognition across various domains, including image and video classification [29,36,37], word embeddings [10,33], recommender systems [34,62,64], audio understanding [24,52], single-cell analysis in biology [31], networks and graphs [60,65,71], and image-text settings [8,26,30,48]. Beyond classification, hyperbolic representations facilitate hierarchical recognition [9,16], learning from limited samples [15,23,41,68], interpretability [19], robustness [35,58], and other tasks. For a comprehensive overview of advances in hyperbolic learning, we refer to recent surveys [45,49]. A common assumption in the literature is that hierarchical information is known and fixed *a priori*. In this work, we flip the perspective to investigate how different hierarchical designs affect hyperbolic embeddings, aiming to potentially enhance integration of hierarchical ontologies in hyperbolic deep learning.

3 Hyperbolic embedding algorithms for hierarchical data

3.1 Preliminaries

Throughout this work, we are given a tree-like data structure $T = (V, E)$, containing a set of nodes V and a set of edges E, with each edge $e \in E$ connecting two vertices. We strive to obtain a continuous analogue of T by embedding each node $v \in V$ in an embedding space, such that the distance between two nodes corresponds one-to-one to the shortest path between the nodes in the tree, as given by the number of edges between them. Let $\phi : V \mapsto \mathbb{D}^n$ denote the embedding function that takes nodes as input and outputs their embedding in an n-dimensional hyperbolic space $\mathbb{D}^n$.

Following [14,46,55], we will operate in the Poincaré ball model of hyperbolic space for the embeddings. For an n-dimensional space, let $(\mathbb{D}^n, \mathfrak{g}^n)$ denote the Riemannian manifold of the Poincaré ball model, given as:

$$\mathbb{D}^n = \big\{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}||^2 < 1\big\}, \qquad \mathfrak{g}^n = \lambda_{\mathbf{x}} I_n, \quad \lambda_{\mathbf{x}} = \frac{2}{1 - ||x||^2}. \tag{1}$$

A key operator for hyperbolic embedding algorithms is the distance between two vectors in hyperbolic space. Here, it denotes the distance between the embeddings of two nodes. For nodes $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{D}^n$, the distance is given as:

$$d_{\mathbb{D}}(\mathbf{v}_1, \mathbf{v}_2) = 2 \tanh^{-1} \big(|| - \mathbf{v}_1 \oplus \mathbf{v}_2||\big), \tag{2}$$

where $\oplus$ denotes the Möbius addition, defined as:

$$\mathbf{v}_1 \oplus \mathbf{v}_2 = \frac{(1 + 2\langle \mathbf{v}_1, \mathbf{v}_2 \rangle + ||\mathbf{v}_2||^2)\mathbf{v}_1 + (1 - ||\mathbf{v}_1||^2)\mathbf{v}_2}{1 + 2\langle \mathbf{v}_1, \mathbf{v}_2 \rangle + ||\mathbf{v}_1||^2||\mathbf{v}_2||^2}. \tag{3}$$

Using this manifold and distance function, we outline below how different algorithms generate hyperbolic embeddings for hierarchical data. We focus on two types of algorithms: general-purpose methods that optimize embeddings via gradient descent and hierarchy-specific approaches that constructively embed trees.

3.2 Gradient-based hyperbolic embeddings

Gradient-based hyperbolic embeddings are general-purpose approaches that take any graph structure as input and yield a hyperbolic embedding of each node, where the embedding uses the edges between nodes as objective. In this work, we investigate two canonical approaches: Poincaré Embeddings [46] and Hyperbolic Entailment Cones [14].

Poincaré Embeddings. In the seminal work of Nickel and Kiela [46], the goal is to embed V using contrastive learning with edges E as positive pairs. Let $\Theta = \{\theta_i\}_{i=1}^{|V|}$ denote the embeddings of nodes in hyperbolic space. The estimation of Θ optimized under the following objective:

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(\Theta), \quad \text{s.t. } \forall \theta_i \in \Theta : ||\theta_i|| < 1. \tag{4}$$

Here, the loss is determined by the edges that connect two nodes. Specifically in the context of tree-like structures, edges denote hypernym-hyponym relations. With D the set of hypernym-hyponym relations the contrastive loss is given as:

$$\mathcal{L}(\Theta) = \sum_{(u,v)\in D} \log \frac{\exp(-d_{\mathbb{D}}(u,v))}{\sum_{v' \in N(u)} \exp(-d_{\mathbb{D}}(u,v'))}, \tag{5}$$

with $N(u)$ denoting the set of nodes not directly connected to u. To optimize Θ, the parameters are initialized as random vectors in a unit ball of dimensionality d and subsequently optimized using gradient descent in hyperbolic space [2,4].

Hyperbolic Entailment Cones. A limitation of the contrastive loss in Poincaré Embeddings is the absence of an explicit objective to preserve hierarchical order. Consequently, nodes deep in the hierarchy may be placed near the origin, reducing the utility of their embeddings. To address this, Ganea et al. [14] reinterpret hierarchical relations as partial orderings defined by cones in hyperbolic space. They extend the contrastive loss to a max-margin variant, aiming for each parent node u to encapsulate its child nodes v. Specifically, each child v must fall within the entailment cone of its parent u. The loss is defined as:

$$\mathcal{L} = \sum_{(u,v)\in D} E(u,v) + \sum_{(u',v')\in A\setminus D} \max(0, \gamma - E(u',v')), \tag{6}$$

with A the set of all node pairs, γ a margin, and the energy loss given as:

$$E(u,v) = \max(0, \Xi(u,v) - \psi(u)), \tag{7}$$

where $\psi(u)$ denotes the aperture of the cone based on its root point u, or equivalently: the size of the entailment cone. The closer u is to the origin, the larger its aperture, reflecting the intuition that points near the origin correspond to higher levels in the tree structure. Lastly, $\Xi(u,v)$ measures the angle between u and v. If v has a higher norm and its angle relative to u is smaller than the aperture of u, the embedding is considered correct, and no loss is incurred. Otherwise, the loss scales with the angular error. Similar to Poincaré Embeddings, this objective can be directly optimized using gradient descent in hyperbolic space. In practice, nodes are initialized with Poincaré Embeddings and refined using the Hyperbolic Entailment Cones objective.

Algorithm 1 Construction-based hyperbolic tree embeddings [55]

1: **Input:** Tree $T = (V, E)$, scaling factor $\tau > 0$ and root node v_1 with $\phi(v_1) = \mathbf{0}$.
2: **for** $v \in V$ **do**
3: Isometrically reflect $\phi(v)$ to the origin and apply the same to its parent.
4: Generate $\mathbf{x}_1, \ldots, \mathbf{x}_{\deg(v)}$ uniformly distributed points on a unit hypersphere.
5: Rotate the points such that $\mathbf{x}_1$ is aligned with the reflected parent embedding.
6: Scale $\mathbf{x}_1, \ldots, \mathbf{x}_{\deg(v)}$ according to τ and the tree distance to v.
7: Reflect rotated and scaled points back.
8: **end for**

3.3 Construction-based hyperbolic embeddings

Gradient-based approaches are general-purpose and operate on a wide range of graphs, including those that are not strictly acyclic or have nodes with multiple inheritance, as in the case of Poincaré Embeddings [46]. However, this versatility often comes at the expense of embedding quality, with the resulting hyperbolic embeddings Θ of the nodes V retaining only partial information from the original graph. In contrast, construction-based methods [55,56] embed trees directly, sacrificing flexibility in the types of graphs they can handle in favor of producing high-quality embeddings that preserve nearly all the original tree structure.

The general approach of construction-based methods is outlined in Algorithm 1. These methods embed a root node at the origin and iteratively traverse the tree, positioning each child node on a sphere centered around its parent. This approach offers strong theoretical guarantees for low distortion and is highly efficient, with linear complexity relative to the number of nodes, avoiding complex optimization problems. However, these methods are limited to tree structures and often require arbitrary-precision arithmetic to achieve low-distortion embeddings.

The core distinction among the construction-based hyperbolic embeddings lies in step 4 of Algorithm 1. The distortion of the resulting embedding depends heavily on the degree of separation between the generated points. However, generating an arbitrary number of uniformly separated points on an n-dimensional hypersphere remains an open problem [54]. Sala et al. [55] propose two approaches for generating hyperspherical points at this step. The first involves placing points at the vertices of a hypercube inscribed within the hypersphere, leveraging coding theory [42]. Specifically, they use the *Hadamard* code, enabling the placement of $2^{\lfloor \log_2 n \rfloor}$ points with a fixed pairwise distance. While this method is computationally efficient and produces predictable results, it suffers from poor separation between points, resulting in higher distortion. Additionally, it imposes a strict requirement on the dimension n, namely:

$$2^{\lfloor \log_2 n \rfloor} \geq \deg_{\max}(V). \tag{8}$$

Their second approach involves *precomputing* 1000 hyperspherical points using the method from [39] and sampling from these as needed. This method often results in lower distortion compared to the first approach and offers greater flexibility regarding the dimension n. However, it has drawbacks, including higher variance in results, with problematic outliers for certain trees, and increased computational cost for smaller trees.

In terms of scalability, the Hadamard method is highly efficient, as an $n \times n$ Hadamard matrix can be constructed in $O(\log_2(n))$ time. The precomputed method incurs minimal initial computation but later only requires tensor sampling. Overall, constructive methods are significantly faster than optimization-based methods.

4 Experimental setup

4.1 Data

Hierarchies for controlled experiments. For our first experiment we generate a variety of tree structures using the NetworkX library [21] with $N = 256, 512, 1024$ nodes to evaluate how different hierarchical structures affect the hyperbolic representation learning. The selected trees encompass diverse structural properties and are defined as follows:

- **Full r-ary trees** *(balanced)* We generate full r-ary trees [59] with r values ranging from 2 to 5. In an r-ary tree, all non-leaf nodes have exactly r children and all levels are full except for some rightmost position of the bottom level (if a leaf at the bottom level is missing, then so are all of the leaves to its right, resulting in trees of varying branching factors and depths. Intuitively, higher values for r result in wider trees lower values for deeper trees r.
- **Binomial tree** *(imbalanced)* A binomial tree is constructed iteratively, with each step having twice the number of nodes as the previous step, forming a hierarchical structure. A binomial tree of order k is defined recursively by linking two binomial trees of order $k-1$, where the root of one is the leftmost child of the root of the other. Thus, the tree grows imbalanced.
- **Barabási–Albert tree** *(long-tailed)* The Barabási–Albert graph [1] of n nodes is generated by attaching new nodes each with m edges that are preferentially attached to existing nodes with high degree, namely preferential attachment. We generate Barabási–Albert graphs with $m = 1$, which is guaranteed to form a tree. This trees' degree distribution follows a power-law distribution ($P(k) = k^{-3}$). The resulting tree captures the scale-free nature observed in many real-world networks, where a few nodes have a high degree and most other nodes have a small degree.

These trees are chosen to represent a diverse range of topologies, from balanced and uniform (r-ary trees) to skewed (binomial and Barabási–Albert trees) [40], while simultaneously allowing us to have control over the number of nodes to enable direct comparisons between different hierarchical organizations. They are visualized for $n = 16$ in Figure 1.

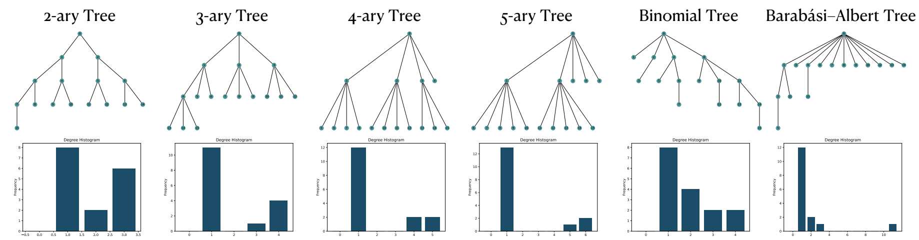

Fig. 1. The different hierarchies and their degree histograms used in our experiments. We investigate various balanced (r-ary trees) and imbalanced trees (binomial and Barabási-Albert trees) to help us understand which dimensions of hierarchy design are most important for their hyperbolic embedding.

Real-world use cases. To investigate the potential impact of our findings in real-world scenarios, we use the ImageNet[7] [4] (based on WordNet[12]) and Pizza [5] ontologies as case studies, demonstrating the influence of alternative ontology designs on the quality of hyperbolic embeddings. The original ontologies, comprising 1778 and 100 nodes respectively, contain multiple inheritance and are non-tree structures. For each ontology, we construct a single inheritance version by applying the DFS algorithm to extract a spanning tree [61]. To minimize tree height while preserving the number of nodes, we restructure the hierarchy by merging the children of parent nodes into siblings. This reduces the height of ImageNet hierarchy from 13 to 8 and Pizza hierarchy from 7 to 5 in the reorganized ontologies.

4.2 Implementation details

In our experiments, we analyze the impact of varying embedding dimensions. For the controlled experiment with generated trees, we use embedding dimensions of $d = 10$, $d = 20$, and $d = 130$. Since the Hadamard method encodes a tree with a minimum dimension d determined by Equation 8, we set the embedding dimensions to $d = 40$ for the Pizza ontology and $d = 70$ for the ImageNet ontology, corresponding to their maximum degrees of 23 and 39, respectively.

Each embedding algorithm is configured using its recommended hyperparameter settings from the corresponding papers. For Poincaré embeddings, we adopt the settings used in the WordNet nouns experiment. Specifically, we use a constant learning rate of 1, with an initial burn-in phase of 20 epochs at a reduced learning rate of 0.1. Training continues for a total of $10,000$ epochs, with a batch size of 50. For each positive example, we randomly sample 50 negative examples.

For the entailment cones method, following Ganea et al. [14], we first pretrain using Poincaré embeddings for 100 epochs using a learning rate of 5.0 and a burn-in learning rate of 0.5 for the initial 20 epochs. Subsequently, we train with

[4] https://observablehq.com/@mbostock/imagenet-hierarchy
[5] https://protege.stanford.edu/ontologies/pizza/pizza.owl

entailment cones loss for 300 epochs using a learning rate of 1.0. Both pretraining and training use a batch size equal to the number of nodes, meaning each epoch consists of a single step. We observed that increasing the number of steps led to overfitting, which adversely affected some metrics. For the construction-based approaches, following Sala et al. [55], the scaling factor τ is set to

$$\tau = \frac{1}{1.3 * \ell} \log\Big(\frac{2 - \frac{\epsilon}{2}}{\frac{\epsilon}{2}}\Big), \tag{9}$$

where ϵ is the machine precision of the applied floating point format and ℓ is the maximum path length of the tree, to avoid numerical problems while still obtaining near optimal results.

4.3 Embedding evaluation metrics

Following the conventions in hyperbolic embedding literature [46,55,56], we focus on three metrics to evaluate the quality of tree embeddings. The first metric is average relative distortion, which measures the average relative embedding error between all pairs of nodes in V, given as follows for $N = |V|$ nodes:

$$D_{avg}(\phi) = \frac{1}{N(N-1)} \sum_{u \neq v} \frac{|d_{\mathbb{D}}(\phi(u), \phi(v)) - d_T(u, v)|}{d_T(u, v)}. \tag{10}$$

This metric measures how much the hyperbolic distance on the embeddings differs from the tree distance between all node pairs. The second metric is worst-case distortion, which specifically measures the ratio between the largest stretching and shrinking factor of pairwise distances:

$$D_{wc}(\phi) = \max_{u \neq v} \frac{d_{\mathbb{D}}(\phi(u), \phi(v))}{d_T(u, v)} \Big(\min_{u \neq v} \frac{d_{\mathbb{D}}(\phi(u), \phi(v))}{d_T(u, v)} \Big)^{-1}. \tag{11}$$

Where the average distortion measures the global distortion, the worst-case distortion captures large local distortions. The third metric is the mean average precision (MAP), given here as:

$$\mathrm{MAP}(\phi) = \frac{1}{N} \sum_{u \in V} \frac{1}{\deg(u)} \sum_{v \in \mathcal{N}_V(u)} \frac{\Big| \mathcal{N}_V(u) \cap \phi^{-1}\Big(B_{\mathbb{D}}(u, v)\Big)\Big|}{\Big|\phi^{-1}\Big(B_{\mathbb{D}}(u, v)\Big)\Big|}, \tag{12}$$

with $\deg(u)$ the degree of node u, $\mathcal{N}_V(u)$ the neighboring nodes of u, and $B_{\mathbb{D}}(u, v) \subset \mathbb{D}^n$ a closed ball centered at the embedding $\phi(u)$ of u with hyperbolic radius $d_{\mathbb{D}}(\phi(u), \phi(v))$. Intuitively, the MAP is a reconstruction measure, which identifies how well we can find back neighboring nodes in the area surrounding each embedded node.

5 Experiments

For our experiments, we focus on four research questions to explore key aspects of embedding hierarchies: (i) width versus depth, (ii) balanced versus imbalanced structures, (iii) few versus many nodes, and (iv) few versus many embedding dimensions. We address each question sequentially. Finally, we apply the insights gained to a case study, where we revisit the ImageNet and Pizza hierarchies and propose an alternative organization that improves hyperbolic embeddings.

5.1 Is it better to design deep or wide hierarchies?

In the first experiment, we address a fundamental question: given the same number of nodes, should hierarchies be designed to be deep or wide? r-ary trees, as visualized in Figure 2, provide an ideal structure for this investigation. Depth versus width inherently carries semantic implications for hierarchical organization. Our objective is to quantify its impact on the resulting hyperbolic embeddings.

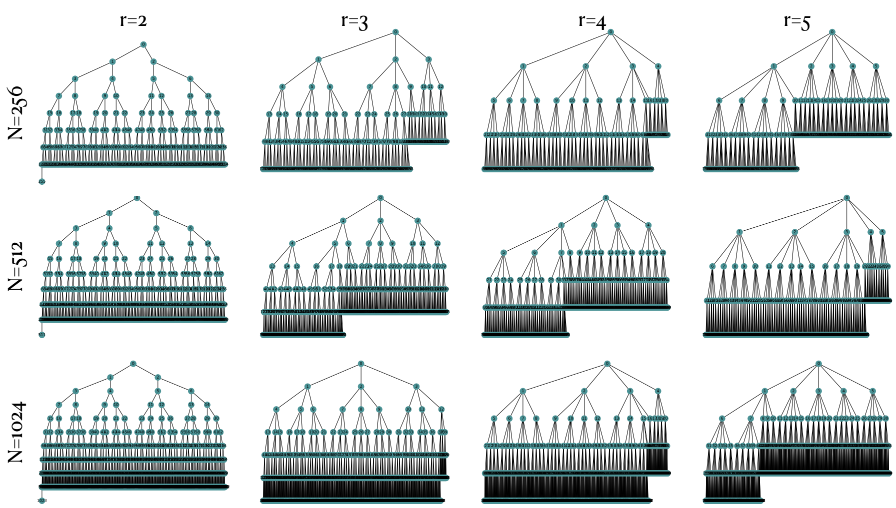

Fig. 2. Visualizing depth versus width in r-ary trees. We show hierarchies for 256, 512, and 1024 nodes for four branching factors, ranging from 2 to 5. The higher the branching factor, the wider the tree and the fewer hierarchical layers that are required to reach the same number of nodes in the hierarchy.

Figure 3 presents the average distortion, worst-case distortion, and MAP as functions of r for all r-ary trees. As the branching factor r increases, the hierarchy becomes wider, requiring fewer layers to reach the same number of nodes. For Poincaré embeddings, wider hierarchies result in higher distortions because the algorithm relies solely on contrastive learning, ignoring the partial order between hierarchical layers. Thus, it performs the worst among the approaches. Poincaré embeddings with $r = 2$ achieve lower distortion and higher MAP than the entailment method at high branching factors, effectively capturing deeper hierarchies.

In contrast to Poincaré, all other hyperbolic embedding methods exhibit the opposite trend: wider hierarchies improve embedding quality, particularly reducing average distortion. Construction-based methods outperform Poincaré with $r = 2$ in both metrics, highlighting a key trade-off. These methods achieve the

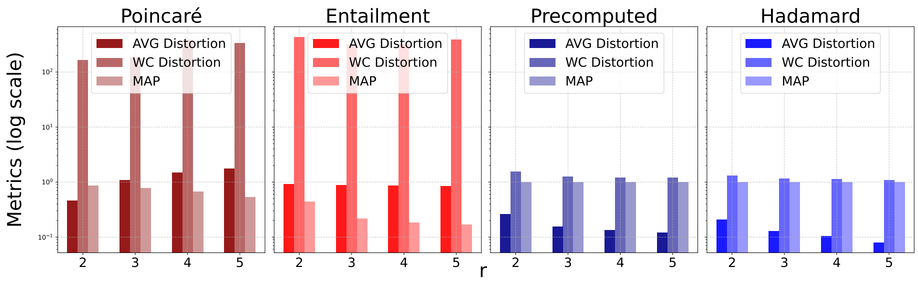

Fig. 3. Investigating depth versus width across four hyperbolic embedding algorithms, using r-ary trees with r ranging from 2 to 5, with hierarchies of 512 nodes and 20 embedding dimensions. For all methods except Poincaré embeddings, we find that wide and shallow hierarchies lead to lower distortion than thin and deep hierarchies.

best overall scores. Thus, we recommend using construction-based algorithms paired with wide hierarchies to achieve optimal hyperbolic embeddings.

5.2 What is the impact of hierarchy imbalance?

In the second experiment, we examine the impact of hierarchy imbalance on embedding performance. Real-world hierarchies naturally tend to follow long-tailed distributions and power laws [13]. However, current embedding algorithms are agnostic to imbalance, and its effect on distortion remains largely unexplored. To address this, we compare two imbalanced hierarchies—based on binomial distributions and the Barabási–Albert model with four balanced r-ary trees. The results for all algorithms and evaluation metrics are presented in Table 1.

Table 1. Balanced versus imbalanced hyperbolic embeddings. We show the results across 4 embedding methods, 6 hierarchies, and 3 evaluation metrics with 512 nodes and 20 embedding dimensions. BA denotes the hierarchy from the Barabási–Albert construction, which can't be performed for the Hadamard construction due to having a maximum degree of 86 (Equation 8), hence the dash. For each algorithm, we highlight the best and worst score over the hierarchies. Overall, we find that balance is not critical for hyperbolic embedding. It is better to have a wide imbalanced hierarchy than a deep balanced hierarchy.

	Gradient-based						**Construction-based**					
	Poincaré			**Entailment**			**Precomputed**			**Hadamard**		
	D_{avg}	D_{wc}	MAP	D_{avg}	D_{wc}	MAP	D_{avg}	D_{wc}	MAP	D_{avg}	D_{wc}	MAP
Balanced												
2-ary	0.459	164.777	0.866	0.914	434.177	0.439	0.259	1.539	1	0.207	1.297	1
3-ary	1.085	183.974	0.770	0.878	316.338	0.217	0.156	1.252	1	0.127	1.155	1
4-ary	1.471	390.397	0.671	0.855	323.967	0.183	0.133	1.201	1	0.103	1.121	1
5-ary	1.770	336.711	0.534	0.837	383.626	0.169	0.120	1.201	1	0.080	1.092	1
Imbalanced												
Binomial	1.439	69.530	0.171	0.863	224.731	0.304	0.249	1.542	1	0.186	1.257	1
BA	2.791	3607.95	0.020	0.802	731.914	0.231	0.140	1.329	1	-	-	-

Interestingly, imbalanced hierarchies do not necessarily yield the highest distortion. Focusing on the two construction-based methods, which outperform the others overall, we observe that the binomial and Barabási-Albert trees perform competitively with the r-ary trees. Their distortion is consistently better than the 2-ary trees but worse than the 5-ary trees. In summary, a wide imbalanced hierarchy is preferable to a deep balanced hierarchy, indicating that enforcing hierarchical balance is not a strict requirement. However, the best results are achieved when hierarchies are both wide and balanced, as imbalance increases depth.

5.3 What is the impact of more nodes on embedding quality?

The larger the hierarchy, the deeper the knowledge it represents, but this also increases the complexity of the corresponding embedding. In the third experiment, we examine how the average distortion of various hierarchies changes as a function of the number of nodes. Table 2 presents results for hierarchies with 256, 512, and 1024 nodes. Table 5 reports the MAPs of the same experiments.

Table 2. The effect of the number of nodes on average distortion across all hyperbolic embedding algorithms. BA represents the hierarchy generated by the Barabási–Albert model. For Poincaré embeddings, increasing the number of nodes reduces distortion, as more node pairs are available for contrastive learning. In contrast, for other methods, adding more nodes slightly increases distortion due to the added complexity from greater depth.

	Gradient-based						**Construction-based**					
	Poincaré			**Entailment**			**Precomputed**			**Hadamard**		
	256	512	1024	256	512	1024	256	512	1024	256	512	1024
Balanced												
2-ary	0.880	0.459	0.229	0.816	0.914	0.960	0.220	0.259	0.300	0.176	0.207	0.240
3-ary	1.439	1.085	0.752	0.742	0.878	0.940	0.124	0.156	0.160	0.102	0.127	0.130
4-ary	2.129	1.471	1.092	0.695	0.855	0.928	0.102	0.133	0.137	0.079	0.103	0.105
5-ary	2.472	1.770	1.385	0.657	0.837	0.919	0.115	0.120	0.156	0.078	0.080	0.103
Imbalanced												
Binomial	1.736	1.439	0.988	0.717	0.863	0.932	0.207	0.249	0.298	0.161	0.186	0.211
BA	3.444	2.791	2.206	0.595	0.802	0.903	0.108	0.140	0.178	-	-	-

Notably, for Poincaré embeddings, larger hierarchies reduce distortion. This is due to the contrastive learning objective, which benefits from more node pairs, improving optimization. For all other methods, results remain largely stable, with a slight positive correlation between the number of nodes and distortion. This outcome relates to the findings of the first experiment: larger hierarchies tend to be deeper, and embedding algorithms that incorporate partial order perform better on shallower hierarchies. We conclude that for most embedding algorithms, enriching hierarchies with more nodes only slightly increases distortion, highlighting that a strong increase in semantic complexity has minimal impact on embedding quality.

5.4 How many embedding dimensions are sufficient?

Hyperbolic geometry enables representation learning in compact spaces [45]. In the fourth experiment, we analyze the effect of embedding dimensionality on average distortion across different algorithms and hierarchies. The results, shown in Table 3, indicate that all approaches perform better with fewer embedding dimensions. Notably, hyperbolic entailment cones and the Hadamard construction are largely agnostic to dimensionality, exhibiting consistent performance across all dimensions. These findings align with existing literature on the efficiency of hyperbolic geometry in lower-dimensional settings.

Table 3. The effect of embedding dimensionality d across all four embedding algorithms for average distortion, with all hierarchies using 512 nodes. Hyperbolic geometry allows for embedding in low-dimensional spaces. Across all algorithms, using fewer dimensions does not hamper performance, and can even lead to better scores for Poincaré embeddings and the precomputed construction-based approach.

	Gradient-based						**Construction-based**					
	Poincaré			**Entailment**			**Precomputed**			**Hadamard**		
	10	20	130	10	20	130	10	20	130	10	20	130
Balanced												
2-ary	0.365	0.459	0.461	0.915	0.914	0.914	0.151	0.259	0.842	0.207	0.207	0.207
3-ary	1.130	1.085	1.105	0.880	0.878	0.878	0.095	0.156	0.467	0.127	0.127	0.127
4-ary	1.487	1.471	1.458	0.857	0.855	0.854	0.087	0.133	0.374	0.103	0.103	0.103
5-ary	1.838	1.770	1.834	0.840	0.837	0.836	0.088	0.120	0.325	0.080	0.080	0.080
Imbalanced												
Binomial	1.435	1.439	1.435	0.865	0.863	0.861	0.118	0.249	0.682	-	0.186	0.186
BA	2.801	2.791	2.784	0.805	0.802	0.802	0.109	0.140	0.371	-	-	0.114

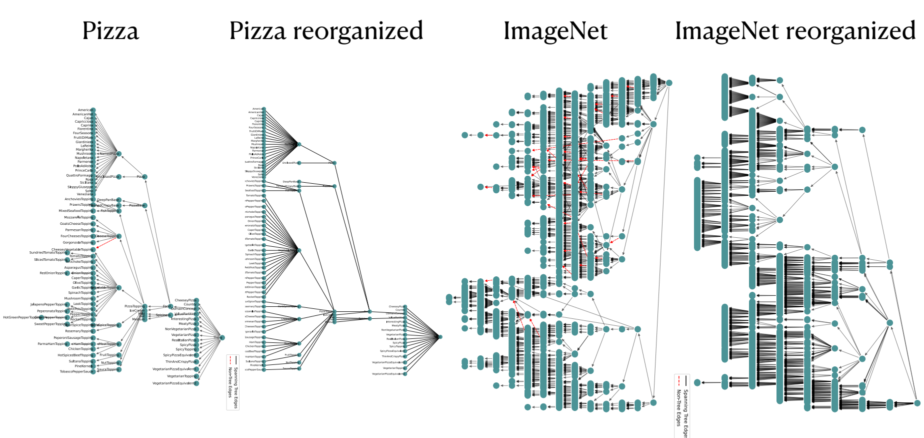

Fig. 4. Original and reorganized real-world ontologies Red edges indicate non-tree edges. Reorganization removes multiple inheritance and reduces tree height.

5.5 Case study: The Pizza and ImageNet ontologies

Lastly, we analyze the practical impact of our recommendation—favoring width over height—on real-world ontologies and evaluate the effects of multiple inheritance. The Pizza and ImageNet ontologies provide suitable case studies, as shown in Figure 4. The figure illustrates the original ontologies alongside their single inheritance and reorganized versions, which have been adjusted to reduce height while maintaining the same number of nodes.
Table 4 summarizes the results for the original, single inheritance, and reorganized versions of the ontologies.

Table 4. The effect of hierarchy re-organization on their hyperbolic embedding. To showcase our own recommendations, we take the existing ImageNet and Pizza hierarchies. Both have edges that create multiple inheritance, rendering them unusable for three out of four methods. Removing multiple inheritance allows for the use of more effective hyperbolic embedders. By including our other lessons as well, we arrive at a re-organization that leads to vastly better distortion and MAP scores.

	Gradient-based						**Construction-based**					
	Poincaré			**Entailment**			**Precomputed**			**Hadamard**		
	D_{avg}	D_{wc}	MAP	D_{avg}	D_{wc}	MAP	D_{avg}	D_{wc}	MAP	D_{avg}	D_{wc}	MAP
Pizza												
Original	3.321	7066.671	0.059	-	-	-	-	-	-	-	-	-
+ single inheritance	3.387	10509.346	0.051	0.499	511.594	0.195	0.234	1.538	1	0.126	1.180	1
+ reorganized	3.422	9343.566	0.045	0.452	1454.972	0.164	0.167	1.329	1	**0.089**	**1.118**	**1**
ImageNet												
Original	0.809	3983.563	0.087	-	-	-	-	-	-	-	-	-
+ single inheritance	0.722	2745.952	0.220	0.961	2364.827	0.293	0.725	885.622	0.725	0.297	1.647	1
+ reorganized	1.008	12715.625	0.156	0.955	4096.000	0.164	0.507	2.698	1	**0.171**	**1.232**	**1**

Notably, Poincaré embeddings are the only method applicable to the original multiple inheritance graphs, as other, more effective methods are incompatible with such structures. The results clearly demonstrate that reorganizing the ontologies significantly improves the average distortion across all methods except for Poincaré. Interestingly, for Poincaré embeddings, deeper hierarchies perform better, whereas for all other methods, wider hierarchies yield superior results.

However, there is a trade-off between expressivity and the structural adjustments made to reduce hierarchy depth. Enforcing a wider and less deep hierarchy can result in the loss of certain semantics, which may diminish the ontology's expressivity. For example, in the Pizza ontology, *Cheese Vegetable Topping* is both a child of *Cheese Topping* and *Vegetable Topping*, but one of these relationships is removed during the single inheritance process. Similarly, in the reorganized hierarchy, *Cheese Burger* which is originally connected through *Hamburger* to *Sandwich*, are flattened to become direct children of *Sandwich*. These adjustments improve downstream performance, particularly for construction-based methods. If preserving semantic expressivity is more critical, deeper hierarchies and Poincaré embeddings may be preferable. These findings highlight a nuanced trade-off: wider hierarchies generally optimize hyperbolic embeddings for performance, they may not be ideal for tasks requiring more semantics.

6 Recommendations

We offer the following recommendations for ontology engineers when designing ontologies or knowledge graph schemas for use with hyperbolic embeddings:

- **Design hierarchies for width:** The most effective embedding algorithms leverage the hierarchical order between nodes to generate embeddings. Consequently, these algorithms perform best with wide hierarchies that have high branching factors, rather than deep narrow trees with slower branching.
- **Do not worry about balance:** Current algorithms are largely agnostic to the balance between subtrees. Interestingly, our findings indicate that when balance is not prioritized or feasible, embedding performance is not significantly impacted. It is better to have a wide imbalanced hierarchy than a deep balanced hierarchy. Achieving both high width and balance leads to the best performance.
- **Hyperbolic embeddings can handle additional node complexity:** We find that a significant increase in the number of nodes only moderately impacts distortion. While more complex data structures lead to more challenging embedding optimization, strong enforcement of node sparsity is not required to maintain effective embeddings.
- **Avoid multiple inheritance:** While Poincaré embeddings can handle hierarchies with multiple inheritance, high-performance embedding algorithms do not support them. Therefore, to minimize distortion, it is best to have single inheritance. This approach is also recommended in many current ontology evaluation methodologies.

These recommendations should be seen as augmenting the main aim of ontology design which is to reflect the domain for the identified task. Hence, these recommendations serve to help ontology engineers balance the need to reflect the domain and the resulting ontologies effectiveness for use in downstream tasks.

7 Conclusion

In this work, we shed light on the relationship between hierarchy design and hyperbolic embeddings. Current hyperbolic literature assumes that hierarchies are fixed prior knowledge and focuses on minimizing embedding distortion. Here, we take the opposite approach by empirically investigating how different design choices can help improve hyperbolic embeddings. In the future, we plan to move beyond structural features of hierarchies to incorporate semantic aspects, including the ontology languages used and the information embedded in labels or literals. This could be done by introducing additional loss functions to balance distortion with semantic constraints. If ontology details (e.g. labels) are available, we could employ them via, e.g., embeddings. We hope that this study encourages future work in how knowledge graphs and data can be designed from the outset to improve down-stream machine learning performance.

8 Acknowledgments

This work was partially supported by the EU's Horizon Europe research and innovation programme within the ENEXA project (grant Agreement no. 101070305).

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. science **286**(5439), 509–512 (1999)
2. Becigneul, G., Ganea, O.E.: Riemannian adaptive optimization methods. In: International Conference on Learning Representations (2018)
3. Beydoun, G., Lopez-Lorca, A.A., García-Sánchez, F., Martínez-Béjar, R.: How do we measure and improve the quality of a hierarchical ontology? Journal of Systems and Software **84**(12), 2363–2373 (Dec 2011). https://doi.org/10.1016/j.jss.2011.07.010, https://linkinghub.elsevier.com/retrieve/pii/S0164121211001853
4. Bonnabel, S.: Stochastic gradient descent on riemannian manifolds. IEEE Transactions on Automatic Control **58**(9), 2217–2229 (2013)
5. Bourgaux, C., Guimarães, R., Koudijs, R., Lacerda, V., Ozaki, A.: Knowledge Base Embeddings: Semantics and Theoretical Properties. In: Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning. pp. 823–833 (8 2024). https://doi.org/10.24963/kr.2024/77, https://doi.org/10.24963/kr.2024/77
6. Cao, Z., Xu, Q., Yang, Z., Cao, X., Huang, Q.: Geometry Interaction Knowledge Graph Embeddings. Proceedings of the AAAI Conference on Artificial Intelligence **36**(5), 5521–5529 (Jun 2022). https://doi.org/10.1609/aaai.v36i5.20491, https://ojs.aaai.org/index.php/AAAI/article/view/20491
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. Desai, K., Nickel, M., Rajpurohit, T., Johnson, J., Vedantam, S.R.: Hyperbolic image-text representations. In: International Conference on Machine Learning. pp. 7694–7731. PMLR (2023)
9. Dhall, A., Makarova, A., Ganea, O., Pavllo, D., Greeff, M., Krause, A.: Hierarchical image classification using entailment cone embeddings. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 836–837 (2020)
10. Dhingra, B., Shallue, C., Norouzi, M., Dai, A., Dahl, G.: Embedding text in hyperbolic spaces. In: Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing. pp. 59–69 (2018)
11. Dong, J., Zhang, Q., Huang, X., Duan, K., Tan, Q., Jiang, Z.: Hierarchy-Aware Multi-Hop Question Answering over Knowledge Graphs. In: Proceedings of the ACM Web Conference 2023. pp. 2519–2527. ACM, Austin TX USA (Apr 2023). https://doi.org/10.1145/3543507.3583376, https://dl.acm.org/doi/10.1145/3543507.3583376
12. Fellbaum, C.: Wordnet. In: Theory and applications of ontology: computer applications, pp. 231–243. Springer (2010)
13. Fix, B.: Hierarchy and the power-law income distribution tail. Journal of Computational Social Science **1**(2), 471–491 (2018)
14. Ganea, O., Bécigneul, G., Hofmann, T.: Hyperbolic entailment cones for learning hierarchical embeddings. In: International conference on machine learning. pp. 1646–1655. PMLR (2018)
15. Gao, Z., Wu, Y., Jia, Y., Harandi, M.: Curvature generation in curved spaces for few-shot learning. In: International Conference on Computer Vision (2021)
16. Ghadimi Atigh, M., Schoep, J., Acar, E., Van Noord, N., Mettes, P.: Hyperbolic image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4453–4462 (2022)

17. Guarino, N., Welty, C.: Evaluating ontological decisions with OntoClean. Communications of the ACM **45**(2), 61–65 (Feb 2002). https://doi.org/10.1145/503124.503150, https://dl.acm.org/doi/10.1145/503124.503150
18. Guizzardi, G., Fonseca, C.M., Almeida, J.P.A., Sales, T.P., Benevides, A.B., Porello, D.: Types and taxonomic structures in conceptual modeling: A novel ontological theory and engineering support. Data & Knowledge Engineering **134**, 101891 (Jul 2021). https://doi.org/10.1016/j.datak.2021.101891, https://linkinghub.elsevier.com/retrieve/pii/S0169023X21000185
19. Gulshad, S., Long, T., van Noord, N.: Hierarchical explanations for video action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3703–3708 (2023)
20. Gmez-Prez, A.: Evaluation of ontologies. International Journal of Intelligent Systems **16**(3), 391–409 (Mar 2001). https://doi.org/10.1002/1098-111X(200103)16:3<391::AID-INT1014>3.0.CO;2-2, https://onlinelibrary.wiley.com/doi/10.1002/1098-111X(200103)16:3<391::AID-INT1014>3.0.CO;2-2
21. Hagberg, A., Swart, P.J., Schult, D.A.: Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Laboratory (LANL), Los Alamos, NM (United States) (2008)
22. Hofer, M., Obraczka, D., Saeedi, A., Köpcke, H., Rahm, E.: Construction of Knowledge Graphs: Current State and Challenges. Information **15**(8), 509 (Aug 2024). https://doi.org/10.3390/info15080509, https://www.mdpi.com/2078-2489/15/8/509
23. Hong, J., Fang, P., Li, W., Han, J., Petersson, L., Harandi, M.: Curved geometric networks for visual anomaly recognition. IEEE Transactions on Neural Networks and Learning Systems (2023)
24. Hong, J., Hayder, Z., Han, J., Fang, P., Harandi, M., Petersson, L.: Hyperbolic audio-visual zero-shot learning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7873–7883 (2023)
25. Huang, Z., Chiang, M.F., Lee, W.C.: LinE: Logical Query Reasoning over Hierarchical Knowledge Graphs. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 615–625. ACM, Washington DC USA (Aug 2022). https://doi.org/10.1145/3534678.3539338, https://dl.acm.org/doi/10.1145/3534678.3539338
26. Ibrahimi, S., Atigh, M.G., Van Noord, N., Mettes, P., Worring, M.: Intriguing properties of hyperbolic embeddings in vision-language models. Transactions on Machine Learning Research (2024)
27. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. IEEE Transactions on Neural Networks and Learning Systems **33**(2), 494–514 (Feb 2022). https://doi.org/10.1109/TNNLS.2021.3070843, https://ieeexplore.ieee.org/document/9416312/
28. Kendall, E.F., McGuinness, D.L.: Ontology Engineering. Synthesis Lectures on Data, Semantics, and Knowledge, Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-031-79486-5, https://link.springer.com/10.1007/978-3-031-79486-5
29. Kim, S., Jeong, B., Kwak, S.: Hier: Metric learning beyond class labels via hierarchical regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19903–19912 (2023)
30. Kim, W., Chun, S., Kim, T., Han, D., Yun, S.: Hype: Hyperbolic entailment filtering for underspecified images and texts. In: European Conference on Computer Vision. pp. 247–265. Springer (2025)

31. Klimovskaia, A., Lopez-Paz, D., Bottou, L., Nickel, M.: Poincaré maps for analyzing complex hierarchies in single-cell data. Nature communications **11**(1), 2966 (2020)
32. Law, M., Liao, R., Snell, J., Zemel, R.: Lorentzian distance learning for hyperbolic representations. In: International Conference on Machine Learning. pp. 3672–3681. PMLR (2019)
33. Le, M., Roller, S., Papaxanthos, L., Kiela, D., Nickel, M.: Inferring concept hierarchies from text corpora via hyperbolic embeddings. arXiv preprint arXiv:1902.00913 (2019)
34. Li, A., Yang, B., Huo, H., Chen, H., Xu, G., Wang, Z.: Hyperbolic neural collaborative recommender. IEEE Transactions on Knowledge and Data Engineering **35**(9), 9114–9127 (2022)
35. Li, H., Chen, Z., Xu, Y., Hu, J.: Hyperbolic anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17511–17520 (2024)
36. Li, Y.L., Wu, X., Liu, X., Wang, Z., Dou, Y., Ji, Y., Zhang, J., Li, Y., Lu, X., Tan, J., et al.: From isolated islands to pangea: Unifying semantic space for human action understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16582–16592 (2024)
37. Liu, S., Chen, J., Pan, L., Ngo, C.W., Chua, T.S., Jiang, Y.G.: Hyperbolic visual embedding learning for zero-shot recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9273–9281 (2020)
38. Long, T., Mettes, P., Shen, H.T., Snoek, C.G.: Searching for actions on the hyperbole. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1141–1150 (2020)
39. Lovisolo, L., Da Silva, E.: Uniform distribution of points on a hyper-sphere with applications to vector bit-plane encoding. IEE Proceedings-Vision, Image and Signal Processing **148**(3), 187–193 (2001)
40. Łuczak, T., Magner, A., Szpankowski, W.: Asymmetry and structural information in preferential attachment graphs. Random Structures & Algorithms **55**(3), 696–718 (2019)
41. Ma, R., Fang, P., Drummond, T., Harandi, M.: Adaptive poincaré point to set distance for few-shot classification. In: AAAI Conference on Artificial Intelligence (2022)
42. MacWilliams, Sloane: The theory of error-correcting codes. Elsevier Science Publishers BV google schola **2**, 39–47 (1977)
43. McDaniel, M., Storey, V.C.: Evaluating Domain Ontologies: Clarification, Classification, and Challenges. ACM Computing Surveys **52**(4), 1–44 (Jul 2020). https://doi.org/10.1145/3329124, https://dl.acm.org/doi/10.1145/3329124
44. McGuinness, D.L.: Ontologies Come of Age. In: Fensel, D., Hendler, J.A., Lieberman, H., Wahlster, W. (eds.) Spinning the Semantic Web, pp. 171–194. The MIT Press (Jan 2003). https://doi.org/10.7551/mitpress/6412.003.0008, https://direct.mit.edu/books/book/3817/chapter/125261/Ontologies-Come-of-Age
45. Mettes, P., Ghadimi Atigh, M., Keller-Ressel, M., Gu, J., Yeung, S.: Hyperbolic deep learning in computer vision: A survey. International Journal of Computer Vision pp. 1–25 (2024)
46. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. Advances in neural information processing systems **30** (2017)
47. Nickel, M., Kiela, D.: Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In: International conference on machine learning. pp. 3779–3788. PMLR (2018)

48. Pal, A., van Spengler, M., di Melendugno, G.M.D., Flaborea, A., Galasso, F., Mettes, P.: Compositional entailment learning for hyperbolic vision-language models. arXiv preprint arXiv:2410.06912 (2024)
49. Peng, W., Varanka, T., Mostafa, A., Shi, H., Zhao, G.: Hyperbolic deep neural networks: A survey. IEEE Transactions on pattern analysis and machine intelligence **44**(12), 10023–10044 (2021)
50. Peng, Y., Bonald, T., Alam, M.: Refining wikidata taxonomy using large language models. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. p. 5395–5399. CIKM '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3627673.3679156, https://doi.org/10.1145/3627673.3679156
51. Pernisch, R., Dell'Aglio, D., Bernstein, A.: Beware of the hierarchy — An analysis of ontology evolution and the materialisation impact for biomedical ontologies. Journal of Web Semantics **70**, 100658 (Jul 2021). https://doi.org/10.1016/j.websem.2021.100658, https://linkinghub.elsevier.com/retrieve/pii/S1570826821000330
52. Petermann, D., Wichern, G., Subramanian, A., Le Roux, J.: Hyperbolic audio source separation. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
53. Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., García-Castro, R.: LOT: An industrial oriented ontology engineering framework. Engineering Applications of Artificial Intelligence **111**, 104755 (May 2022). https://doi.org/10.1016/j.engappai.2022.104755, https://linkinghub.elsevier.com/retrieve/pii/S0952197622000525
54. Saff, E.B., Kuijlaars, A.B.: Distributing many points on a sphere. The mathematical intelligencer **19**, 5–11 (1997)
55. Sala, F., De Sa, C., Gu, A., Ré, C.: Representation tradeoffs for hyperbolic embeddings. In: International conference on machine learning. pp. 4460–4469. PMLR (2018)
56. Sarkar, R.: Low distortion delaunay embedding of trees in hyperbolic plane. In: International symposium on graph drawing. pp. 355–366. Springer (2011)
57. Shimizu, C., Hammar, K., Hitzler, P.: Modular ontology modeling. Semantic Web **14**(3), 459–489 (Apr 2023). https://doi.org/10.3233/SW-222886, https://journals.sagepub.com/doi/full/10.3233/SW-222886
58. van Spengler, M., Zahálka, J., Mettes, P.: Adversarial attacks on hyperbolic networks. In: European Conference on Computer Vision Workshops (2024)
59. Storer, J.A.: An introduction to data structures and algorithms. Springer Science & Business Media (2012)
60. Sun, L., Zhang, Z., Zhang, J., Wang, F., Du, Y., Su, S., Yu, P.S.: Perfect: A hyperbolic embedding for joint social network alignment. In: IEEE International Conference on Data Mining (2020)
61. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM journal on computing **1**(2), 146–160 (1972)
62. Tu, L., Meng, S., Qi, L., Xu, X., Zhang, X.: Hyperbolic contrastive learning with second order sampling for collaborative filtering. In: 2024 IEEE International Conference on Web Services (ICWS). pp. 281–290. IEEE (2024)
63. Weikum, G., Dong, X.L., Razniewski, S., Suchanek, F.: Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. Foundations and Trends® in Databases **10**(2-4), 108–490 (Jul 2021). https://doi.org/10.1561/1900000064, https://www.nowpublishers.com/article/Details/DBS-064

64. Yang, M., Li, Z., Zhou, M., Liu, J., King, I.: Hicf: Hyperbolic informative collaborative filtering. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2212–2221 (2022)
65. Yang, Y., Wu, L., Zhang, K., Hong, R., Zhou, H., Zhang, Z., Zhou, J., Wang, M.: Hyperbolic graph learning for social recommendation. IEEE Transactions on Knowledge and Data Engineering (2023)
66. Yu, T., Liu, T.J., Tseng, A., De Sa, C.: Shadow cones: Unveiling partial orders in hyperbolic space. arXiv preprint arXiv:2305.15215 (2023)
67. Yu, Z., Nguyen, T., Gal, Y., Ju, L., Chandra, S.S., Zhang, L., Bonnington, P., Mar, V., Wang, Z., Ge, Z.: Skin lesion recognition with class-hierarchy regularized hyperbolic embeddings. In: International conference on medical image computing and computer-assisted intervention. pp. 594–603. Springer (2022)
68. Zhang, B., Jiang, H., Feng, S., Li, X., Ye, Y., Ye, R.: Hyperbolic knowledge transfer with class hierarchy for few-shot learning. In: IJCAI. pp. 3723–3729 (2022)
69. Zhang, Y., Pietrasik, M., Xu, W., Reformat, M.: Hierarchical Topic Modelling for Knowledge Graphs. In: Groth, P., Vidal, M.E., Suchanek, F., Szekley, P., Kapanipathi, P., Pesquita, C., Skaf-Molli, H., Tamper, M. (eds.) The Semantic Web, vol. 13261, pp. 270–286. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-06981-9_16, https://link.springer.com/10.1007/978-3-031-06981-9_16
70. Zhang, Z., Cai, J., Zhang, Y., Wang, J.: Learning Hierarchy-Aware Knowledge Graph Embeddings for Link Prediction. Proceedings of the AAAI Conference on Artificial Intelligence **34**(03), 3065–3072 (Apr 2020). https://doi.org/10.1609/aaai.v34i03.5701, https://ojs.aaai.org/index.php/AAAI/article/view/5701
71. Zhou, M., Yang, M., Xiong, B., Xiong, H., King, I.: Hyperbolic graph neural networks: A tutorial on methods and applications. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 5843–5844 (2023)

9 Appendix

9.1 What is the impact of more nodes on embedding reconstruction?

In Section 5.3, we investigate the effect of the number of nodes on average distortion. Table 5 presents the MAP values for all hyperbolic embedding algorithms applied to trees with 256, 512, and 1024 nodes. The results show that construction-based methods achieve the highest MAP values, with the exception of the Barabási–Albert construction, which cannot be applied to the Hadamard method due to the maximum tree degree constraint (Equation 8). For gradient-based methods, increasing the number of nodes generally leads to a slight reduction in MAP values. We conclude that for gradient-based algorithms, enriching hierarchies with additional nodes only marginally increases distortion, while construction-based methods remain unaffected. This highlights that even a significant increase in semantic complexity has minimal impact on embedding quality.

Table 5. The effect of the number of nodes on MAPs across all hyperbolic embedding algorithms. BA represents the hierarchy generated by the Barabási–Albert model. For gradient-based methods, increasing the number of nodes in most cases reduces MAP, whereas for construction-based methods, all experiments result in the maximum MAP of 1.

	Gradient-based						**Construction-based**					
	Poincaré			**Entailment**			**Precomputed**			**Hadamard**		
	256	512	1024	256	512	1024	256	512	1024	256	512	1024
Balanced												
2-ary	0.949	0.866	0.791	0.404	0.439	0.397	1	1	1	1	1	1
3-ary	0.862	0.770	0.620	0.254	0.217	0.206	1	1	1	1	1	1
4-ary	0.798	0.671	0.566	0.219	0.183	0.158	1	1	1	1	1	1
5-ary	0.650	0.534	0.509	0.205	0.169	0.136	1	1	1	1	1	1
Imbalanced												
Binomial	0.169	0.171	0.154	0.342	0.304	0.251	1	1	1	1	1	1
BA	0.028	0.020	0.0136	0.249	0.231	0.192	1	1	1	-	-	-