Representation Tradeoffs for Hyperbolic Embeddings

Frederic Sala [1] **Christopher De Sa** [2] **Albert Gu** [1] **Christopher Ré** [1]

Abstract

Hyperbolic embeddings offer excellent quality with few dimensions when embedding hierarchical data structures. We give a combinatorial construction that embeds trees into hyperbolic space with arbitrarily low distortion without optimization. On WordNet, this algorithm obtains a mean-average-precision of 0.989 with only two dimensions, outperforming existing work by 0.11 points. We provide bounds characterizing the precision-dimensionality tradeoff inherent in any hyperbolic embedding. To embed general metric spaces, we propose a hyperbolic generalization of multidimensional scaling (h-MDS). We show how to perform exact recovery of hyperbolic points from distances, provide a perturbation analysis, and give a recovery result that enables us to reduce dimensionality. Finally, we extract lessons from the algorithms and theory above to design a scalable PyTorch-based implementation that can handle incomplete information.

1. Introduction

Recently, hyperbolic embeddings have been proposed as a way to capture hierarchy information for network and natural language processing tasks (Nickel & Kiela, 2017; Chamberlain et al., 2017). This approach is an exciting way to fuse structural information (for example, from knowledge graphs or synonym hierarchies) with the continuous representations favored by modern machine learning methods.

To understand the intuition behind hyperbolic embeddings' superior capacity, note that trees can be embedded with arbitrarily low distortion into the Poincaré disk, a two-dimensional model of hyperbolic space (Sarkar, 2011). In contrast, Bourgain's theorem (Linial et al., 1995) shows that Euclidean space cannot achieve comparably low distortion for trees—even using an unbounded number of dimensions.

Many graphs, such as complex networks (Krioukov et al., 2010), the Internet (Krioukov et al., 2009), and social networks (Verbeek & Suri, 2016), are known to have tree-like or hyperbolic structure and thus befit hyperbolic embeddings. Indeed, recent works show that hyperbolic representations are suitable for many hierarchies (e.g. the question answering (Q/A) system HyperQA in Tay et al. (2018), vertex classifiers in Chamberlain et al. (2017), and link prediction (Nickel & Kiela, 2017)). However, the optimization problems underlying the embedding techniques in these works are challenging, motivating us to seek fundamental insights and to understand the subtle tradeoffs involved.

We begin by considering the case where we are given an input graph that is a tree or nearly tree-like, and our goal is to produce a low-dimensional hyperbolic embedding that preserves all distances. This leads to a simple combinatorial strategy that directly places points instead of minimizing a surrogate loss function. It is both fast (nearly linear time) and has formal quality guarantees. The approach proceeds in two phases: we (1) produce an embedding of a graph into a weighted tree, and (2) embed that tree into the hyperbolic disk. In particular, we consider an extension of an elegant embedding of trees into the Poincaré disk by Sarkar (2011) and work on low-distortion graph embeddings into tree metrics (Abraham et al., 2007). For trees, this approach has nearly perfect quality. On the WordNet hypernym graph reconstruction, it obtains a nearly perfect mean average precision (MAP) of 0.989 using just 2 dimensions. The best published numbers for WordNet in Nickel & Kiela (2017) range between 0.823 and 0.87 for 5 to 200 dimensions.

We analyze this construction to extract fundamental tradeoffs. One tradeoff involves the embedding dimension, the properties of the graph, and the number of bits of precision used to represent components of embedded points—an important hidden cost. We show that for a fixed precision, the dimension required scales linearly with the length of the longest path. On the other hand, the dimension scales logarithmically with the maximum degree of the tree. This suggests that hyperbolic embeddings should have high quality on hierarchies like WordNet but require large dimensions or high precision on graphs with long chains.

To understand how hyperbolic embeddings perform for met-

[1]Department of Computer Science, Stanford University [2]Department of Computer Science, Cornell University. Correspondence to: Frederic Sala <fredsala@stanford.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

rics that are far from tree-like, we consider a more general problem: given a matrix of distances that arise from points that are embeddable in hyperbolic space of dimension d (not necessarily from a graph), find a set of points that produces these distances. In Euclidean space, the problem is known as multidimensional scaling (MDS) and is solvable using PCA. A key step is a transformation that effectively centers the points, without knowledge of their exact coordinates. It is not obvious how to center points in hyperbolic space, which is curved. We show that in hyperbolic space, a centering operation is still possible with respect to a non-standard mean. In turn, this allows us to reduce the hyperbolic MDS problem (h-MDS) to a standard eigenvalue problem that can be solved with power methods. We also extend classical PCA perturbation analysis (Sibson, 1978; 1979). When applied to distances from graphs induced by real data, h-MDS obtains low distortion on far from tree-like graphs. However, we observe that these solutions may require high precision, which is not surprising in light of our previous analysis.

Finally, we handle increasing amounts of noise in the model, leading naturally into new SGD-based formulations. Like in traditional PCA, the underlying problem is nonconvex. In contrast to PCA, there are local minima that are not global minima—an additional challenge. Our main technical result is that an SGD-based algorithm initialized with an h-MDS solution can recover the submanifold the data is on—even in some cases in which the data is perturbed by noise that can be full dimensional. Our algorithm essentially provides new recovery results for convergence of Principal Geodesic Analysis (PGA) in hyperbolic space. We implemented the resulting SGD-based algorithm using PyTorch. Finally, we note that all of our algorithms can handle incomplete distance information through standard techniques.

2. Background

We provide intuition connecting hyperbolic space and tree distances, discuss the metrics used to measure embedding fidelity, and discuss the relationship between the reconstruction and learning problems for graph embeddings.

Hyperbolic spaces The Poincaré disk $\mathbb{H}_2$ is a two-dimensional model of hyperbolic geometry with points located in the interior of the unit disk, as shown in Figure 1. A natural generalization of $\mathbb{H}_2$ is the Poincaré ball $\mathbb{H}_r$, with elements inside the unit ball. The Poincaré models offer several useful properties, chief among which is mapping conformally to Euclidean space. That is, angles are preserved between hyperbolic and Euclidean space. Distances, on the other hand, are not preserved, but are given by

$$d_H(x,y) = \operatorname{acosh}\left(1 + 2\frac{\|x-y\|^2}{(1-\|x\|^2)(1-\|y\|^2)}\right).$$

There are some potentially unexpected consequences of this formula, and a simple example gives intuition about a key technical property that allows hyperbolic space to embed trees. Consider three points inside the unit disk: the origin 0, and points x and y with $\|x\| = \|y\| = t$ for some $t > 0$. As shown on the right of Figure 1, as $t \to 1$ (i.e., the points move towards the outside of the disk), in flat Euclidean space, the ratio $\frac{d_E(x,y)}{d_E(x,0)+d_E(0,y)}$ is *constant* with respect to t (blue curve). In contrast, the ratio $\frac{d_H(x,y)}{d_H(x,0)+d_H(0,y)}$ approaches 1, or, equivalently, the distance $d_H(x,y)$ approaches $d_H(x,0) + d_H(0,y)$ (red and pink curves). That is, the shortest path between x and y is almost the same as the path through the origin. This is analogous to the property of trees in which the shortest path between two sibling nodes is the path through their parent. This tree-like nature of hyperbolic space is the key property exploited by embeddings. Moreover, this property holds for arbitrarily small angles between x and y.

Lines and geodesics There are two types of geodesics (shortest paths) in the Poincaré disk model: segments of circles that are orthogonal to the disk surface, and disk diameters (Brannan et al., 2012). Our algorithms and proofs make use of a simple geometric fact: *isometric* reflection across geodesics (preserving hyperbolic distances) is represented in this Euclidean model as a *circle inversion*.

Embeddings and fidelity measures An *embedding* is a mapping $f : U \to V$ for spaces U, V with distances d_U, d_V. We measure the quality of embeddings with several *fidelity measures*, presented here from most local to most global.

Recent work (Nickel & Kiela, 2017) proposes using the *mean average precision* (MAP). For a graph $G = (V, E)$, let $a \in V$ have neighborhood $\mathcal{N}_a = \{b_1, b_2, \ldots, b_{\deg(a)}\}$, where $\deg(a)$ denotes the degree of a. In the embedding f, consider the points closest to $f(a)$, and define R_{a,b_i} to be the smallest set of such points that contains b_i (that is, R_{a,b_i} is the smallest set of nearest points required to retrieve the ith neighbor of a in f). Then, the MAP is defined to be

$$\text{MAP}(f) = \frac{1}{|V|}\sum_{a\in V}\frac{1}{\deg(a)}\sum_{i=1}^{|\mathcal{N}_a|}\frac{|\mathcal{N}_a \cap R_{a,b_i}|}{|R_{a,b_i}|}.$$

We have $\text{MAP}(f) \le 1$, with 1 as the best case. MAP is not concerned with explicit distances, but only *ranks* between the distances of immediate neighbors. It is a *local* metric.

The standard metric for graph embeddings is distortion D. For an n point embedding,

$$D(f) = \frac{1}{\binom{n}{2}}\left(\sum_{u,v\in U: u\neq v}\frac{|d_V(f(u),f(v)) - d_U(u,v)|}{d_U(u,v)}\right).$$

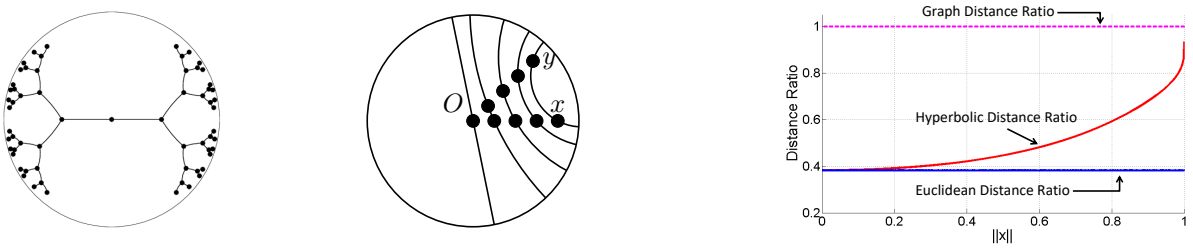

Figure 1. Left: Embedding of a binary tree in the Poincaré disk. Right: Geodesics and distances. As x and y move towards the outside of the disk (i.e., letting $\|x\|, \|y\| \to 1$), the distance $d_H(x, y)$ approaches $d_H(x, O) + d_H(O, y)$.

The best distortion is $D(f) = 0$, preserving the edge lengths exactly. This is a *global* metric, as it depends directly on the underlying distances rather than the local relationships between distances. A variant is the worst-case distortion D_{wc}, defined by

$$D_{\text{wc}}(f) = \frac{\max_{u,v \in U: u \neq v} d_V(f(u), f(v))/d_U(u, v)}{\min_{u,v \in U: u \neq v} d_V(f(u), f(v))/d_U(u, v)}.$$

That is, the wost-case distortion is the ratio of the maximal expansion and the minimal contraction of distances. Note that scaling the unit distance does not affect D_{wc}. The best worst-case distortion is $D_{\text{wc}}(f) = 1$.

Reconstruction and learning If we lack a full set of distances, we can either use the triangle inequality to recover the missing distances, or we can access the scaled Euclidean distances (the inside of the acosh in $d_H(x, y)$), and apply standard matrix completion techniques (Candes & Tao, 2010). Then we compute an embedding using any of the approaches discussed in this paper. We quantify the error introduced by this process experimentally in Section 5.

3. Combinatorial Constructions

We first focus on hyperbolic tree embeddings—a natural approach considering the tree-like behavior of hyperbolic space. We review the embedding of Sarkar (2011). We then provide novel analysis on the precision, revealing fundamental limits of hyperbolic embeddings. In particular, we characterize the bits of precision needed for hyperbolic representations. We extend the construction to r dimensions, and propose to use Steiner nodes to better embed general graphs as trees, building on Abraham et al. (2007).

Embedding trees The nature of hyperbolic space lends itself towards excellent tree embeddings. In fact, it is possible to embed trees into the Poincaré disk $\mathbb{H}_2$ with arbitrarily low distortion (Sarkar, 2011). Remarkably, trees cannot be embedded into Euclidean space with arbitrarily low distortion for *any* number of dimensions. These notions motivate the following two-step process for embedding hierarchies into hyperbolic space: (1) embed the graph $G = (V, E)$ into a tree T, and (2) embed T into the Poincaré ball $\mathbb{H}_d$. We refer to this process as the *combinatorial construction*. Note that we are not required to minimize a loss function. We begin by describing the second stage, where we extend an elegant construction from Sarkar (2011).

3.1. Sarkar's Construction

Algorithm 1 performs an embedding of trees into $\mathbb{H}_2$. The inputs are a scaling factor τ and a node a (of degree $\deg(a)$) from the tree with parent node b. Say a and b have already been embedded into $f(a)$ and $f(b)$ in $\mathbb{H}_2$. The algorithm places the children $c_1, c_2, \ldots, c_{\deg(a)-1}$ into $\mathbb{H}_2$.

A two-step process is used. First, $f(a)$ and $f(b)$ are reflected across a geodesic (using circle inversion) so that $f(a)$ is mapped onto the origin 0 and $f(b)$ is mapped onto some point z. Next, we place the children nodes to vectors $y_1, \ldots, y_{d-1}$ equally spaced around a circle with radius $\frac{e^\tau - 1}{e^\tau + 1}$ (which is a circle of radius τ in the hyperbolic metric), and maximally separated from the reflected parent node embedding z. Lastly, we reflect all of the points back across the geodesic. The isometric properties of reflections imply that all children are now at hyperbolic distance exactly τ from $f(a)$. To embed the entire tree, we place the root at the origin O and its children in a circle around it (as in Step 5 of Algorithm 1), then recursively place their children until all nodes have been placed. This process runs in linear time.

3.2. Analyzing Sarkar's Construction

Sarkar's construction works by separating children sufficiently in hyperbolic space. A key technical idea is to scale all the edges by a factor τ before embedding. We can then recover the original distances by dividing by τ. This transformation exploits the fact that hyperbolic space is not *scale invariant*. Sarkar's construction always captures neighbors perfectly, but Figure 1 implies that increasing the scale preserves the distances between farther nodes better. Indeed, if one sets $\tau = \frac{1+\varepsilon}{\varepsilon}\left(2 \log \frac{\deg_{\max}}{\pi/2}\right)$, then the worst-case distortion D of the resulting embedding is no more than

Algorithm 1 Sarkar's Construction

1: **Input:** Node a with parent b, children to place $c_1, c_2, \ldots, c_{\deg(a)-1}$, partial embedding f containing an embedding for a and b, scaling factor τ
2: $(0, z) \leftarrow \text{reflect}_{f(a)\to 0}(f(a), f(b))$
3: $\theta \leftarrow \arg(z)$ {angle of z from x-axis in the plane}
4: **for** $i \in \{1, \ldots, \deg(a) - 1\}$ **do**
5: $y_i \leftarrow \frac{e^\tau - 1}{e^\tau + 1} \cdot \left(\cos\left(\theta + \frac{2\pi i}{\deg(a)}\right), \sin\left(\theta + \frac{2\pi i}{\deg(a)}\right)\right)$
6: $(f(a), f(b), f(c_1), \ldots, f(c_{\deg(a)-1})) \leftarrow \text{reflect}_{0\to f(a)}(0, z, y_1, \ldots, y_{\deg(x)-1})$
7: **Output:** Embedded $\mathbb{H}_2$ vectors $f(c_1), f(c_2), \ldots, f(c_{\deg(a)-1})$

$1 + \varepsilon$. For trees, Sarkar's construction has arbitrarily high fidelity. However, this comes at a cost: the scaling τ affects the bits of precision required. In fact, we will show that the precision scales logarithmically with the degree of the tree—but linearly with the maximum path length.

How many bits of precision do we need to represent points in $\mathbb{H}_2$? If $x \in \mathbb{H}_2$, then $\|x\| < 1$, so we need sufficiently many bits so that $1 - \|x\|$ will not be rounded to zero. This requires roughly $-\log(1 - \|x\|) = \log \frac{1}{1-\|x\|}$ bits. Say we are embedding two points x, y at distance d. As described in the background, there is an isometric reflection that takes a pair of points (x, y) in $\mathbb{H}_2$ to $(0, z)$ while preserving their distance, so without loss of generality we have that

$$d = d_H(x, y) = d_H(0, z) = \operatorname{acosh}\left(1 + 2\frac{\|z\|^2}{1 - \|z\|^2}\right).$$

Rearranging the terms, we have $(\cosh(d) + 1)/2 = (1 - \|z\|^2)^{-1} \geq (1 - \|z\|)^{-1}/2$. Thus, the number of bits we want so that $1 - \|z\|$ will not be rounded to zero is $\log(\cosh(d)+1)$. Since $\cosh(d) = (\exp(d)+\exp(-d))/2$, this is roughly d bits. That is, in hyperbolic space, we need about d bits to express distances of d (rather than $\log d$ in Euclidean space).[1] This result will be of use below.

Consider the largest distance in the embeddings produced by Algorithm 1. If the longest path length in the tree is ℓ, and each edge has length $\tau = \frac{1}{\varepsilon}\left(2\log\frac{\deg_{\max}}{\pi/2}\right)$, the largest distance is $O(\frac{\ell}{\varepsilon}\log\deg_{\max})$, and we require this number of bits for the representation.

Let us interpret this expression. Note that $\deg_{\max}$ is inside the $\log$ term, so that a bushy tree is not penalized much in precision. On the other hand, the longest path length ℓ is not, so that hyperbolic embeddings struggle with long paths. Moreover, by selecting an explicit graph, we derive a matching lower bound, concluding that to achieve a distortion ε, any construction requires $\Omega\left(\frac{\ell}{\varepsilon}\log(\deg_{\max})\right)$ bits. The argument follows from selecting a graph consisting of $m(\deg_{\max}+1)$ nodes in a tree with a single root and $\deg_{\max}$ chains each of length m (shown in the appendix).

[1] Although it is particularly easy to bound precision in the Poincaré model, this fact holds generally for hyperbolic space independent of model (shown in the appendix).

3.3. Improving the Construction

Our next contribution is a generalization of the construction from the disk $\mathbb{H}_2$ to the ball $\mathbb{H}_r$. Our construction follows the same line as Algorithm 1, but since we have r dimensions, the step where we place children spaced out on a circle around their parent now uses a hypersphere.

Spacing out points on the hypersphere is a classic problem known as *spherical coding* (Conway & Sloane, 1999). As we shall see, the number of children that we can place for a particular angle grows with the dimension. Since the required scaling factor τ gets larger as the angle decreases, we can reduce τ for a particular embedding by increasing the dimension. Note that increasing the dimension helps with bushy trees (large $\deg_{\max}$), but has limited effect on tall trees with small $\deg_{\max}$. We show

Proposition 3.1. *The generalized $\mathbb{H}_r$ combinatorial construction has distortion at most $1 + \varepsilon$ and requires at most $O(\frac{1}{\varepsilon}\frac{\ell}{r}\log\deg_{\max})$ bits to represent a node component for $r \leq (\log\deg_{\max}) + 1$, and $O(\frac{1}{\varepsilon}\ell)$ bits for $r > (\log\deg_{\max}) + 1$.*

To generalize to $\mathbb{H}_r$, we replace Step 5 in Algorithm 1 with a node placement step based on coding theory. The children are placed at the vertices of a hypercube inscribed into the unit hypersphere (and then scaled by τ). Each component of a hypercube vertex has the form $\frac{\pm 1}{\sqrt{r}}$. We index these points using binary sequences $a \in \{0, 1\}^r$ in the following way: $x_a = \left(\frac{(-1)^{a_1}}{\sqrt{r}}, \frac{(-1)^{a_2}}{\sqrt{r}}, \ldots, \frac{(-1)^{a_r}}{\sqrt{r}}\right)$. We space out the children by controlling the distances by selecting a set of binary sequences a with a prescribed minimum Hamming distance—a binary error-correcting code—and placing the children at the resulting hypercube vertices. We provide more details, including our choice of code in the appendix.

3.4. Embedding into Trees

We revisit the first step of the construction: embedding graphs into trees. There are fundamental limits to how well graphs can be embedded into trees; in general, breaking long cycles inevitably adds distortion, as shown in Figure 2. We are inspired by a measure of this limit, the *δ-4 points condition* introduced in Abraham et al. (2007). A graph on n nodes that satisfies the δ-4 points condition has distortion at most $(1 + \delta)^{c_1 \log n}$ for some constant c_1. This result enables our end-to-end embedding to achieve a distortion of at most $D(f) \leq (1 + \delta)^{c_1 \log n}(1 + \varepsilon)$.

The result in Abraham et al. (2007) builds a tree with *Steiner*

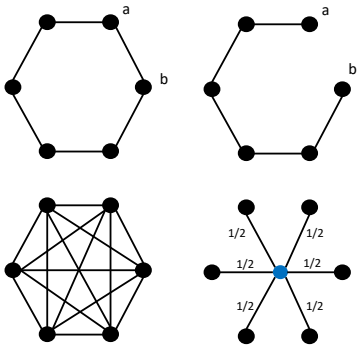

Figure 2. Top: Cycles are an issue in tree embeddings: $d_G(a, b)$ changes from 1 to 5. Bottom: Steiner nodes can help: adding a node and weighting edges maintains the pairwise distances.

nodes. These additional nodes can help control the distances in the resulting weighted tree (Figure 2). Note that Algorithm 1 readily extends to the case of weighted trees.

In summary, the key takeaways of our analysis are:

- There is a fundamental tension between precision and quality in hyperbolic embeddings.
- Hyperbolic embeddings have an exponential advantage in space compared to Euclidean embeddings for short, bushy hierarchies, but will have less of an advantage for graphs that contain long paths.
- Choosing an appropriate scaling factor τ is critical for quality. Later, we will propose to learn this scale factor automatically for computing embeddings in PyTorch.
- Steiner nodes can help improve embeddings of graphs.

4. Hyperbolic Multidimensional Scaling

In this section, we explore a fundamental and more general question than we did in the previous section: if we are given the pairwise distances arising from a set of points in hyperbolic space, can we recover the points? This enables us to produce an embedding for a desired distance metric. The equivalent problem for Euclidean distances is solved with multidimensional scaling (MDS). The goal of this section is to analyze the *hyperbolic MDS* (h-MDS) problem. We describe and overcome the additional technical challenges imposed by hyperbolic distances, and show that exact recovery is possible and interpretable. Afterwards we propose a technique for dimensionality reduction using principal geodesics analysis (PGA) that provides optimization guarantees. In particular, this addresses the shortcomings of h-MDS when recovering points that do not exactly lie on a hyperbolic manifold.

4.1. Exact Hyperbolic MDS

Suppose that there is a set of hyperbolic points $x_1, \ldots, x_n \in \mathbb{H}_r$, embedded in the Poincaré ball and written $X \in \mathbb{R}^{n \times r}$ in matrix form. We observe all the pairwise distances $d_{i,j} = d_H(x_i, x_j)$, but do not observe X: our goal is to use the observed $d_{i,j}$'s to recover X (or some other set of points with the same pairwise distances $d_{i,j}$).

The MDS algorithm in the Euclidean setting makes an important *centering*[2] assumption: the points have mean 0. If an exact embedding for the distances exists, it can be recovered from a matrix factorization. In other words, Euclidean MDS always recovers a centered embedding.

In hyperbolic space, the same algorithm does not work, but we show that it is possible to find an embedding centered at a different mean. More precisely, we introduce a new mean which we call the *pseudo-Euclidean mean*, that behaves like the Euclidean mean in that it enables recovery through matrix factorization. Once the points are recovered in hyperbolic space, they can be recentered around a more canonical mean by translating it to the origin.

Algorithm 2 is our complete algorithm, and for the remainder of this section we will describe how and why it works. We first describe the *hyperboloid model*, an alternate but equivalent model of hyperbolic geometry in which h-MDS is simpler. Of course, we can easily convert between the hyperboloid model and the Poincaré ball model. Next, we show how to reduce the problem to a standard PCA problem, which recovers an embedding centered at the points' pseudo-Euclidean mean. Finally, we discuss the meaning and implications of centering and prove that the algorithm preserves submanifolds as well—that is, if there is an exact embedding in $k < r$ dimensions centered at their canonical mean, then our algorithm will recover it.

The hyperboloid model Define Q to be the diagonal matrix in $\mathbb{R}^{r+1}$ where $Q_{00} = 1$ and $Q_{ii} = -1$ for $i > 0$. For a vector $x \in \mathbb{R}^{r+1}$, $x^T Q x$ is called the *Minkowski quadratic form*. The hyperboloid model is defined as

$$\mathbb{M}_r = \left\{ x \in \mathbb{R}^{r+1} \middle| x^T Q x = 1 \wedge x_0 > 0 \right\},$$

which is endowed with a distance measure $d_H(x, y) = \operatorname{acosh}(x^T Q y)$. For convenience, for $x \in \mathbb{M}_r$ let x_0 denote 0th coordinate $e_0^T x$, and $\vec{x} \in \mathbb{R}^r$ denote the rest of the coordinates[3]. With this notation, the Minkowski bilinear form can be written $x^T Q y = x_0 y_0 - \vec{x}^T \vec{y}$.

[2] We say that points are centered at a particular mean if this mean is at 0. The act of centering refers to applying an isometry that makes the mean of the points 0.

[3] Since $x_0 = \sqrt{1 + \|\vec{x}\|^2}$ is just a function of $\vec{x}$, we can equivalently consider just $\vec{x}$ as being a member of a model of hyperbolic space: This representation is sometimes known as the Gans model.

A new mean Given points $x_1, x_2, \ldots, x_n \in \mathbb{M}_r$ in hyperbolic space, define a variance term

$$\Psi(z; x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} \sinh^2(d_H(x_i, z)).$$

We define a *pseudo-Euclidean mean* to be any local minimum of this expression. Notice that this is independent of any particular model of hyperbolic space, since it is defined only through the hyperbolic distance function d_H.

Lemma 4.1. *Define $X \in \mathbb{R}^{n \times r}$ such that $X^T e_i = \vec{x}_i$ and $u \in \mathbb{R}^n$ such that $u_i = x_{0,i}$. Then*

$$\nabla_{\vec{z}} \Psi(z; x_1, x_2, \ldots, x_n)|_{\vec{z}=0} = -2 \sum_{i=1}^{n} x_{0,i} \vec{x}_i = -2X^T u.$$

This means that 0 is a pseudo-Euclidean mean if and only if $0 = X^T u$. Call some hyperbolic points $x_1, \ldots, x_n$ *pseudo-Euclidean centered* if their average is 0 in this sense: i.e. if $X^T u = 0$. We can always center a set of points without affecting their pairwise distances by simply finding their average, and then sending it to 0 through an isometry.

Recovery via matrix factorization Suppose we observe the pairwise distances $d_H(x_i, x_j)$ of points $x_1, x_2, \ldots, x_n \in \mathbb{M}_r$. This gives the matrix Y such that

$$Y_{i,j} = \cosh\left(d_H(x_i, x_j)\right) = x_{0,i} x_{0,j} - \vec{x_i}^T \vec{x_j}. \quad (1)$$

Defining X and u as in Lemma 4.1, then in matrix form $Y = uu^T - XX^T$. Without loss of generality, suppose that the x_i are centered at their pseudo-Euclidean mean, so that $X^T u = 0$ by Lemma 4.1. This implies that u is an eigenvector of Y with positive eigenvalue, and the rest of Y's eigenvalues are negative. Therefore an eigendecomposition of Y will find $u, \hat{X}$ such that $Y = uu^T - \hat{X}\hat{X}^T$, i.e. it will directly recover X up to rotation.

In fact, running PCA on $-Y = X^T X - uu^T$ to find the n most significant non-negative eigenvectors will recover X up to rotation, and then u can be found by leveraging the fact that $x_0 = \sqrt{1 + \|\vec{x}\|^2}$. This leads to Algorithm 2, with optional post-processing steps for converting the embedding to the Poincaré ball model and for re-centering the points.

A word on centering The MDS algorithm in Euclidean geometry returns points centered at their *Karcher mean* z, which is a point minimizing $\sum d^2(z, x_i)$ (where d is the distance metric). The Karcher center is important for interpreting dimensionality reduction; we use the analogous hyperbolic Karcher mean for PGA in Section 4.2.

Although Algorithm 2 returns points centered at their pseudo-Euclidean mean instead of their Karcher mean, they can be easily recentered by finding their Karcher mean and reflecting it onto the origin. Furthermore, Algorithm 2 preserves the dimension of the embedding:

Algorithm 2

1: **Input: Distance matrix $d_{i,j}$ and rank r**
2: Compute scaled distance matrix $Y_{i,j} = \cosh(d_{i,j})$
3: $X \to \text{PCA}(-Y, r)$
4: Project X from hyperboloid model to Poincaré model: $x \to \frac{x}{1+\sqrt{1+\|x\|^2}}$
5: If desired, center X at a different mean (e.g. the Karcher mean)
6: **return** X

Lemma 4.2. *If a set of points lie in a dimension-k geodesic submanifold, then both their Karcher mean and their pseudo-Euclidean mean lie in the same submanifold.*

This implies that centering with the pseudo-Euclidean mean preserves geodesic submanifolds: If it is possible to embed distances in a dimension-k geodesic submanifold centered and rooted at a Karcher mean, then it is also possible to embed the distances in a dimension-k submanifold centered and rooted at a pseudo-Euclidean mean, and vice versa.

4.2. Reducing Dimensionality with PGA

Given a high-rank embedding (resulting from h-MDS, for example), we may wish to find a lower-rank version. In Euclidean space, one can get the optimal lower rank embedding by simply discarding components. However, this may not be the case in hyperbolic space. Motivated by this, we study dimensionality reduction in hyperbolic space.

As hyperbolic space does not have a linear subspace structure like Euclidean space, we need to define what we mean by lower-dimensional. We follow Principal Geodesic Analysis (Fletcher et al., 2004), (Huckemann et al., 2010). Consider an initial embedding with points $x_1, \ldots, x_n \in \mathbb{H}_2$ and let $d_H : \mathbb{H}_2 \times \mathbb{H}_2 \to \mathbb{R}_+$ be the hyperbolic distance. Suppose we want to map this embedding onto a one-dimensional subspace. (Note that we are considering a two-dimensional embedding and one-dimensional subspace here for simplicity, and these results immediately extend to higher dimensions.) In this case, the goal of PGA is to find a geodesic $\gamma : [0, 1] \to \mathbb{H}_2$ that passes through the mean of the points and that minimizes the squared error (or variance): $f(\gamma) = \sum_{i=1}^{n} \min_{t \in [0,1]} d_H(\gamma(t), x_i)^2$.

This expression can be simplified significantly and reduced to a minimization in Euclidean space. First, we find the mean of the points, the point $\bar{x}$ which minimizes $\sum_{i=1}^{n} d_H(\bar{x}, x_i)^2$.[4] Next, we reflect all the points x_i so that their mean is 0 in the Poincaré disk model; we can

[4]The derivative of the hyperbolic distance has a singularity, that is, $\lim_{y \to x} \partial_x |d_H(x, y)| \to \infty$ for any $x \in \mathbb{H}$. This issue can

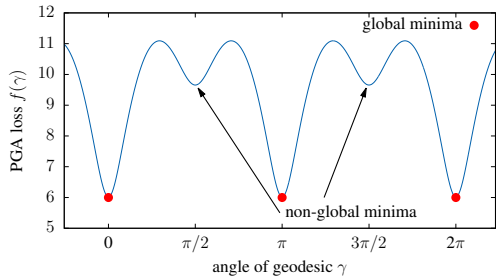

Figure 3. The PGA objective of an example task where the input dataset in the Poincaré disk is $x_1 = (0.8, 0)$, $x_2 = (-0.8, 0)$, $x_3 = (0, 0.7)$ and $x_4 = (0, -0.7)$. Note the presence of non-optimal local minima, unlike PCA.

do this using a circle inversion that maps $\bar{x}$ onto 0 Since reflections are isometric, if γ is a line through 0 and R_γ is the reflection across γ, we have that $d_H(\gamma, x) = \min_{t\in[0,1]} d_H(\gamma(t), x) = \frac{1}{2} d_H(R_l x, x)$.

Combining this with the Euclidean reflection formula and the hyperbolic metric produces

$$f(\gamma) = \frac{1}{4}\sum_{i=1}^{n} \operatorname{acosh}^2\left(1 + \frac{8 d_E(\gamma, x_i)^2}{(1 - \|x_i\|^2)^2}\right),$$

in which d_E is the Euclidean distance from a point to a line. If we define $w_i = \sqrt{8} x_i / (1 - \|x_i\|^2)$ this reduces to the simplified expression $f(\gamma) = \frac{1}{4}\sum_{i=1}^{n} \operatorname{acosh}^2\left(1 + d_E(\gamma, w_i)^2\right)$.

Notice that *the loss function is not convex*. We observe that there can be multiple local minima that are attractive and stable, in contrast to PCA. Figure 3 illustrates this nonconvexity on a simple dataset in $\mathbb{H}_2$ with only four examples. This makes globally optimizing the objective difficult.

Nevertheless, there will always be a region Ω containing a global optimum γ^* that is convex and admits an efficient projection, and where f is convex when restricted to Ω. Thus it is possible to build a gradient descent-based algorithm to recover lower-dimensional subspaces: for example, we built a simple optimizer in PyTorch. We also give a sufficient condition on the data for f above to be convex.

Lemma 4.3. *For hyperbolic PGA if for all i,*

$$\operatorname{acosh}^2\left(1 + d_E(\gamma, w_i)^2\right) < \min\left(1, \frac{1}{3}\|w_i\|^2\right)$$

then f is locally convex at γ.

be mitigated by minimizing d_H^2, which does have a continuous derivative throughout $\mathbb{H}$. The use of $d_H(x, y)$ is a minor instability in Nickel & Kiela (2017); Chamberlain et al. (2017)'s formulation, necessitating guarding against NANs. We discuss this further in the appendix.

Table 1. Dataset statistics.

Dataset	Nodes	Edges	Comment
Bal. Tree	40	39	Tree
Phy. Tree	344	343	Tree
CS PhDs	1025	1043	Tree-like
WordNet	74374	75834	Tree-like
Diseases	516	1188	Dense
Gr-QC	4158	13428	Dense

Table 2. MAP measure for WordNet embedding compared to values in Nickel & Kiela (2017). Closer to 1 is better.

Dataset	C-$\mathbb{H}_2$	FB $\mathbb{H}_5$	FB $\mathbb{H}_{200}$
WordNet	0.989	0.823*	0.87*

As a result, if we initialize in and optimize over a region that contains γ^* and where the condition of Lemma 4.3 holds, then gradient descent will be guaranteed to converge to γ^*. We can turn this result around and read it as a recovery result: if the noise is bounded in this regime, then we are able to provably recover the correct low-dimensional embedding.

5. Experiments

We evaluate the proposed approaches and compare against existing methods. We hypothesize that for tree-like data, the combinatorial construction offers the best performance. For general data, we expect h-MDS to produce the lowest distortion, while it may have low MAP due to precision limitations. We anticipate that dimension is a critical factor (outside of the combinatorial construction). In the appendix, we report on additional datasets, combinatorial construction parameters, and the effect of hyperparameters.

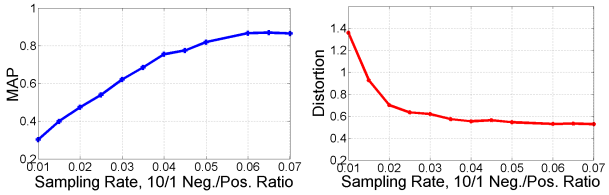

Figure 4. Learning from incomplete information. The distance matrix is sampled, completed, and embedded.

Datasets We consider trees, tree-like hierarchies, and graphs that are not tree-like. Trees include fully-balanced and phylogenetic trees expressing genetic heritage (Hofbauer et al., 2016), available at Sanderson et al. (1994). Nearly tree-like hierarchies include the WordNet hypernym graph (the largest connected component from Nickel & Kiela (2017)) and a graph of Ph.D. advisor-advisee relationships (De Nooy et al., 2011). Also included are datasets

Table 3. Combinatorial and h-MDS techniques, compared against PCA and results from Nickel & Kiela (2017) (asterisks). Left (Distortion): Closer to 0 is better. Right (MAP): Closer to 1 is better.

Dataset	C-$\mathbb{H}_2$	FB $\mathbb{H}_2$	h-MDS	PT	PWS	PCA	FB	C-$\mathbb{H}_2$	FB $\mathbb{H}_2$	h-MDS	PT	PWS	PCA	FB
Bal. Tree	**0.013**	0.425	**0.077**	0.034	0.020	0.496	0.236	**1.0**	0.846	**1.0**	**1.0**	**1.0**	**1.0**	0.859
Phy. Tree	**0.006**	0.832	**0.039**	0.237	0.092	0.746	0.583	**1.0**	0.718	0.675	0.951	0.998	**1.0**	0.811
CS PhDs	**0.286**	0.542	**0.149**	0.298	0.187	0.708	0.336	**0.991**	0.567	0.463	0.799	**0.945**	0.541	0.78
Diseases	**0.147**	0.410	0.111	**0.080**	0.108	0.595	0.764	**0.822**	0.788	0.949	0.995	0.897	**0.999**	0.934
Gr-QC	**0.354**	-	0.530	**0.125**	0.134	0.546	-	0.696	-	0.710	0.733	0.504	0.738	**0.999***

Table 4. Precision and recall for WordNet entity-relationship-entity triple hyperbolic embeddings using combinatorial construction.

Relationship	Precision	Recall
'has instance'	99.97	99.98
'part of'	100.00	99.64
'domain region'	99.66	99.93

that vary in their tree nearness, such as disease relationships (Goh et al., 2007) and protein interactions (Jeong et al., 2001), both available from Rossi & Ahmed (2015). We also include the general relativity and quantum cosmology (Gr-QC) arXiv collaboration network (Leskovec et al., 2007).

Approaches Combinatorial embeddings into $\mathbb{H}_2$ use the $\varepsilon = 0.1$ precision setting; others are considered in the Appendix. We performed h-MDS in floating point precision. We include results for our PyTorch implementation (PT) of an SGD-based algorithm (described later), and a warm start version (PWS) initialized with the high-dimensional combinatorial construction. We compare against classical MDS (i.e., PCA), and the optimization-based approach Nickel & Kiela (2017), which we call FB. The experiments for h-MDS, PyTorch SGD, PCA, and FB used dimensions of 2,5,10,50,100,200; we recorded the best resulting MAP and distortion. Due to the large scale, we did not replicate the best FB numbers on large graphs (i.e., Gr-QC and WordNet); we report their best published MAP numbers (their work does not report distortion). These entries are marked with an asterisk. For the WordNet graph, FB uses the transitive closure; a weighted version of the graph captures the ancestor relationships. The full details are in appendix.

Quality In Table 3 (left), we report the distortion. As expected, for tree or tree-like graphs, the combinatorial construction has exceedingly low distortion. Because h-MDS is meant to recover points exactly, we hypothesized that h-MDS would offer very low distortion on these datasets. Table 3 confirms this: among h-MDS, PCA, and FB, h-MDS consistently offers the lowest distortion, producing, for example, a distortion of 0.039 on the phylogenetic tree. We observe that floating point h-MDS struggles with MAP. We separately confirmed that this is due to precision (by using a high-precision solver). The optimization-based approach is bolstered by appropriate initialization from the combinatorial construction.

Table 3 (right) reports the MAP measure (we additionally include WordNet results in Table 2), which is a local measure. We confirm that the combinatorial construction performs well for tree-like hierarchies, where MAP is close to 1. The construction improves on approaches such as FB that rely on optimization. On larger graphs like WordNet, our approach yields a MAP of 0.989—while their WordNet MAP result is 0.870 at 200 dimensions. This is exciting, as our approach is deterministic and linear-time.

A refined understanding of hyperbolic embeddings may be used to improve the quality and runtime of extant algorithms. Indeed, we embedded WordNet entity-relationship-entity triples (Socher et al., 2013) using the combinatorial construction in 10 dimensions, accurately preserving relationship knowledge (Table 4). This suggests that hyperbolic embeddings are effective at compressing knowledge and may useful for knowledge base completion and Q/A tasks.

SGD-Based Algorithm We built an SGD-based algorithm implemented in PyTorch. The loss function is equivalent to the PGA loss, and so is continuously differentiable.

To evaluate our algorithm's ability to deal with incomplete information, we sample the distance matrix at a ratio of non-edges to edges at $10:1$ following Nickel & Kiela (2017). In Figure 4, we recover a good solution for the phylogenetic tree with a small fraction of the entries; for example, we sampled approximately 4% of the graph for a MAP of 0.74 and distortion of 0.6. We also considered learning the scale of the embedding (details in the appendix). Finally, all of our techniques scale to graphs with millions of nodes.

6. Conclusion and Future Work

Hyperbolic embeddings embed hierarchical information with high fidelity and few dimensions. We explored the limits of this approach by describing scalable, high quality algorithms. We hope the techniques here encourage more follow-on work on the exciting techniques of Nickel & Kiela (2017); Chamberlain et al. (2017).

Acknowledgements

Thanks to Alex Ratner and Avner May for helpful discussion and to Beliz Gunel and Sen Wu for assistance with experiments. We gratefully acknowledge the support of DARPA under No. FA87501720095 and FA87501320039, ONR under No. N000141712266, the Moore Foundation, Okawa Research Grant, American Family Insurance, Accenture, Toshiba, the Secure Internet of Things Project, Google, VMware, Qualcomm, Ericsson, Analog Devices, and members of the Stanford DAWN project: Intel, Microsoft, Teradata, and VMware. 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, DOE, NIH, ONR, or the U.S. Government.

References

Abraham, I., Balakrishnan, M., Kuhn, F., Malkhi, D., Ramasubramanian, V., and Talwar, K. Reconstructing approximate tree metrics. In *Proc. of the 26th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 43–52, Portland, Oregon, 2007.

Abu-Ata, M. and Dragan, F. F. Metric tree-like structures in real-world networks: an empirical study. *Networks*, 67 (1):49–68, 2015.

Benedetti, R. and Petronio, C. *Lectures on Hyperbolic Geometry*. Springer, Berlin, Germany, 1992.

Brannan, D., Esplen, M., and Gray, J. *Geometry*. Cambridge University Press, Cambridge, UK, 2012.

Candes, E. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Chamberlain, B. P., Clough, J. R., and Deisenroth, M. P. Neural embeddings of graphs in hyperbolic space. *arXiv preprint, arXiv:1705.10359*, 2017.

Chen, W., Fang, W., Hu, G., and Mahoney, M. W. On the hyperbolicity of small-world and tree-like random graphs. In *Proc. of the International Symposium on Algorithms and Computation (ISAAC) 2012*, pp. 278–288, Taipei, Taiwan, 2012.

Conway, J. and Sloane, N. J. A. *Sphere Packings, Lattices and Groups*. Springer, New York, NY, 1999.

Cvetkovski, A. and Crovella, M. Hyperbolic embedding and routing for dynamic graphs. In *Proc. of the 28th IEEE International Conference on Computer Communications (INFOCOM 2009)*, Rio de Janeiro, Brazil, 2009.

Cvetkovski, A. and Crovella, M. Multidimensional scaling in the poincaré disk. *Applied mathematics & information sciences*, 2016.

De Nooy, W., Mrvar, A., and Batagelj, V. *Exploratory social network analysis with Pajek*, volume 27. Cambridge University Press, 2011.

Eppstein, D. and Goodrich, M. Succinct greedy graph drawing in the hyperbolic plane. In *Proc. of the International Symposium on Graph Drawing (GD 2008)*, pp. 14–25, Heraklion, Greece, 2008.

Fletcher, P., Lu, C., Pizer, S., and Joshi, S. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8): 995–1005, 2004.

Gabrilovich, E. and Markovitch, S. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proc. of The 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, pp. 1606–1611, Hyderabad, India, 2007.

Ganea, O., Bécigneul, G., and Hofmann, T. Hyperbolic neural networks. *arXiv preprint, arXiv:1805.09112*, 2018a.

Ganea, O., Bécigneul, G., and Hofmann, T. Hyperbolic entailment cones for learning hierarchical embeddings. *arXiv preprint, arXiv:1804.01882*, 2018b.

Goh, K., Cusick, M., Valle, D., Childs, B., Vidal, M., and Barabási, A. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21), 2007.

Gromov, M. Hyperbolic groups. In *Essays in group theory*. Springer, 1987.

Gulcehre, C., Denil, M., Malinowski, M., Razavi, A., Pascanu, R., Hermann, K. M., Battaglia, P., Bapst, V., Raposo, D., Santoro, A., and de Freitas, N. Hyperbolic attention networks. *arXiv preprint, arXiv:1805.09786*, 2018.

Hofbauer, W., Forrest, L., Hollingsworth, P., and Hart, M. Preliminary insights from DNA barcoding into the diversity of mosses colonising modern building surfaces. *Bryophyte Diversity and Evolution*, 38(1):1–22, 2016.

Huckemann, S., Hotz, T., and Munk, A. Intrinsic shape analysis: Geodesic PCA for Riemannian manifolds modulo isometric Lie group actions. *Statistica Sinica*, 20(1): 1–58, 2010.

Jenssen, M., Joos, F., and Perkin, W. On kissing numbers and spherical codes in high dimensions. *arXiv preprint, arXiv:1803.02702*, 2018.

Jeong, H., Mason, S., Barabási, A., and Oltvai, Z. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

Kleinberg, J. Authoritative sources in a hyperlinked environment. 1999. URL http://www.cs.cornell.edu/courses/cs685/2002fa/.

Kleinberg, R. Geographic routing using hyperbolic space. In *Proc. of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pp. 1902–1909, Anchorage, Alaska, 2007.

Krioukov, D., Papadopoulos, F., and Boguná, A. V. M. Curvature and temperature of complex networks. *Physical Review E*, 80(035101), 2009.

Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguná, M. Hyperbolic geometry of complex networks. *Physical Review E*, 82(036106), 2010.

Lamping, J. and Rao, R. Laying out and visualizing large trees using a hyperbolic space. In *Proc. of the 7th annual ACM Symposium on User Interface Software and Technology (UIST 94)*, pp. 13–14, Marina del Rey, California, 1994.

Leskovec, J., Kleinberg, J., and Faloutsos, C. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1 (2), 2007.

Linial, N., London, E., and Rabinovich, Y. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

Nickel, M. and Kiela, D. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, 2017.

Pennec, X. Barycentric subspace analysis on manifolds. *Annals of Statistics*, to appear 2017.

Rossi, R. A. and Ahmed, N. K. The network data repository with interactive graph analytics and visualization. In *Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. URL http://networkrepository.com.

Sanderson, M. J., Donoghue, M. J., Piel, W. H., and Eriksson, T. TreeBASE: a prototype database of phylogenetic analyses and an interactive tool for browsing the phylogeny of life. *American Journal of Botany*, 81(6), 1994.

Sarkar, R. Low distortion Delaunay embedding of trees in hyperbolic plane. In *Proc. of the International Symposium on Graph Drawing (GD 2011)*, pp. 355–366, Eindhoven, Netherlands, September 2011.

Sibson, R. Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society, Series B*, 40(2):234–238, 1978.

Sibson, R. Studies in the robustness of multidimensional scaling: Perturbational analysis of classical scaling. *Journal of the Royal Statistical Society, Series B*, 41(2):217–229, 1979.

Socher, R., Chen, D., Manning, C. D., and Ng, A. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pp. 926–934, Lake Tahoe, NV, 2013.

Tay, Y., Tuan, L. A., and Hui, S. C. Hyperbolic representation learning for fast and efficient neural question answering. In *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM 2018)*, pp. 583–591, Los Angeles, California, 2018.

Verbeek, K. and Suri, S. Metric embedding, hyperbolic space, and social networks. *Computational Geometry*, 59 Issue C:1–12, 2016.

Walter, J. A. H-MDS: a new approach for interactive visualization with multidimensional scaling in the hyperbolic space. *Information Systems*, 29(4):273–292, 2004.

Wilson, R., Hancock, E., Pekalska, E., and Duin, R. Spherical and hyperbolic embeddings of data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36 (11):2255–2269, 2014.

Zhang, M. and Fletcher, P. Probabilistic principal geodesic analysis. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, Lake Tahoe, NV, 2013.

A. Glossary of Symbols

Symbol	Used for
x, y, z	vectors in the Poincaré ball model of hyperbolic space
d_H	metric distance between two points in hyperbolic space
d_E	metric distance between two points in Euclidean space
d_U	metric distance between two points in metric space U
d	a particular distance value
$d_{i,j}$	the distance between the ith and jth points in an embedding
$\mathbb{H}_r$	the Poincaré ball model of r-dimensional Hyperbolic space
r	the dimension of a Hyperbolic space
$\mathbb{H}$	Hyperbolic space of an unspecified or arbitrary dimension
$\mathbb{M}_r$	the Minkowski (hyperboloid) model of r-dimensional Hyperbolic space
f	an embedding
$\mathcal{N}_a$	neighborhood around node a in a graph
$R_{a,b}$	the smallest set of closest points to node a in an embedding f that contains node b
MAP(f)	the mean average precision fidelity measure of the embedding f
$D(f)$	the distortion fidelity measure of the embedding f
$D_{\text{wc}}(f)$	the worst-case distortion fidelity measure of the embedding f
G	a graph, typically with node set V and edge set E
T	a tree
a, b, c	nodes in a graph or tree
$\deg(a)$	the degree of node a
$\deg_{\max}$	maximum degree of a node in a graph
ℓ	the longest path length in a graph
τ	the scaling factor of an embedding
$\text{reflect}_{x \to y}$	a reflection of x onto y in hyperbolic space
$\arg(z)$	the angle that the point z in the plane makes with the x-axis
X	matrix of points in hyperbolic space
Y	matrix of transformed distances
γ	geodesic used in PGA
w_i	transformed points used in PGA

Table 5. Glossary of variables and symbols used in this paper.

B. Related Work

Our study of representation tradeoffs for hyperbolic embeddings was motivated by exciting recent approaches towards such embeddings in Nickel & Kiela (2017) and Chamberlain et al. (2017). Earlier efforts proposed using hyperbolic spaces for routing, starting with Kleinberg's work on geographic routing (Kleinberg, 2007). Cvetkovski & Crovella (2009) performed hyperbolic embeddings and routing for dynamic networks. Recognizing that the use of hyperbolic space for routing required a large number of bits to store the vertex coordinates, Eppstein & Goodrich (2008) introduced a scheme for succinct embedding and routing in the hyperbolic plane.

Another very recent effort also proposes using hyperbolic cones (similar to the cones that are the fundamental building block used in Sarkar (2011) and our work) as a heuristic for embedding entailment relations, i.e. directed acyclic graphs (Ganea et al., 2018b). The authors also propose to optimize on the hyperbolic manifold using its exponential map, as opposed to our approach of finding a closed form for the embedding should it exist (Section 4). An interesting avenue for future work is to compare both optimization methods empirically and theoretically, i.e., to understand the types of recovery guarantees under noise that such methods have.

A pair of recent approaches seek to add hyperbolic operations to neural networks. Gulcehre et al. (2018) introduces a hyperbolic version of the attention mechanism using the hyperboloid model of hyperbolic space. In Ganea et al. (2018a), building blocks from certain networks are generalized to operate with Riemannian manifolds.

There have been previous efforts to perform multidimensional scaling in hyperbolic space (the h-MDS problem), often in the context of visualization (Lamping & Rao, 1994). Most propose descent methods in hyperbolic space (e.g. (Cvetkovski & Crovella, 2016), (Walter, 2004)) and fundamentally differ from ours. Arguably the most relevant is Wilson et al. (2014), which mentions exact recovery as an intermediate result, but ultimately suggests a heuristic optimization. Our h-MDS analysis characterizes the recovered embedding and manifold and obtains the correctly centered one—a key issue in MDS. For example, this allows us to properly find the components of maximal variation. Furthermore, we discuss robustness to noise and produce optimization guarantees when a perfect embedding doesn't exist.

Several papers have studied the notion of hyperbolicity of networks, starting with the seminal work on hyperbolic graphs Gromov (1987). More recently, Chen et al. (2012) considered the hyperbolicity of small world graphs and tree-like random graphs. Abu-Ata & Dragan (2015) performed a survey that examines how well real-world networks can be approximated by trees using a variety of tree measures and tree embedding algorithms. To motivate their study of tree metrics, Abraham et al. (2007) computed a measure of tree likeness on a Internet infrastructure network.

We use matrix completion (closure) to perform embeddings with incomplete data. Matrix completion is a celebrated problem. Candes & Tao (2010) derive bounds on the minimum number of entries needed for completion for a fixed rank matrix; they also introduce a convex program for matrix completion operating at near the optimal rate.

Principal geodesic analysis (PGA) generalizes principal components analysis (PCA) for the manifold setting. It was introduced and applied to shape analysis in Fletcher et al. (2004) and extended to a probabilistic setting in Zhang & Fletcher (2013). There are other variants; the geodesic principal components analysis (GPCA) of Huckemann et al. (2010) uses our loss function. A further generalization of PCA to Riemannian manifolds is the Barycentric Subspace Analysis of Pennec (2017), where it is shown that there is no direct and perfect analogue of PCA in negatively curved spaces.

C. Low-Level Formulation Details

Implementations of our algorithms are available at `https://github.com/HazyResearch/hyperbolics`. A few comments are helpful to understand the reformulation. In particular, we simply minimize the squared hyperbolic distance with a learned scale parameter, τ, e.g., :

$$\min_{x_1,\ldots,x_n,\tau} \sum_{1\le i<j\le n} (\tau d_H(x_i,x_j) - d_{i,j})^2$$

We typically require that $\tau \ge 0.1$.

- On continuity of the derivative of the loss: Note that

$$\partial_x \mathsf{acosh}(1+x) = \frac{1}{\sqrt{(1+x)^2-1}} = \frac{1}{\sqrt{x(x+2)}} \text{ hence } \lim_{x\to 0} \partial_x \mathsf{acosh}(1+x) = \infty.$$

  Thus, $\lim_{y\to x} \partial_x d_H(x,y) = \infty$. In particular, if two points happen to get near to one another during execution, gradient-based optimization becomes unstable. Note that $\exp\{\mathrm{acosh}(1+\mathrm{x})\}$ suffers from a similar issue, and is used in both (Nickel & Kiela, 2017; Chamberlain et al., 2017). This change may increase numerical instability, and the public code for these approaches does indeed take steps like masking out updates to mitigate NANs. In contrast, the following may be more stable:

$$\partial_x \mathsf{acosh}(1+x)^2 = 2\frac{\mathsf{acosh}(1+x)}{\sqrt{x(x+2)}} \text{ and in particular } \lim_{x\to 0} \partial_x \mathsf{acosh}(1+x)^2 = 2$$

  The limits follows by simply applying L'Hopital's rule. In turn, this implies the square formulation is continuously differentiable. Note that it is not convex.

- One challenge is to make sure the gradient computed by PyTorch has the appropriate curvature correction (the Riemannian metric), as is well explained by Nickel & Kiela (2017). The modification is straightforward: we create a subclass of NN.PARAMETER called HYPERBOLIC_PARAMETER. This wrapper class allows us to walk the tree to apply the appropriate correction to the metric (which amounts to multiplying $\nabla_w f(w)$ by $\frac{1}{4}(1-\|w\|^2)^2$. After calling the BACKWARD function, we call a routine to walk the autodiff tree to find such parameters and correct them. This allows HYPERBOLIC_PARAMETER and traditional parameters to be freely mixed.

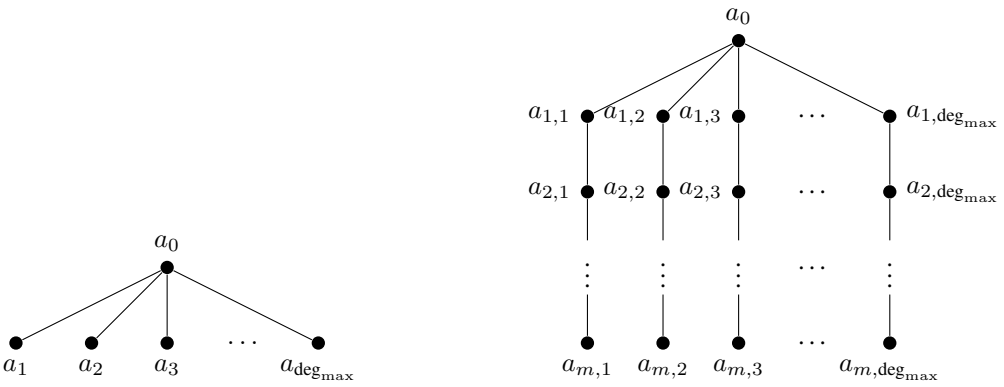

Figure 5. Explicit graphs G_m used to derive precision lower bound. Left: $m = 1$ case (star graph). Right: $m > 1$.

- We project back on the hypercube following Nickel & Kiela (2017) and use gradient clipping with bounds of $[-10^5, 10^5]$. This allows larger batch sizes to more fully utilize the GPU.

D. Combinatorial Construction Proofs

Precision vs. model We first provide a simple justification of the fact (used in Section 3.2) that representing distances d requires about d bits in hyperbolic space—independent of the model of the space. Formally, we show that the number of bits needed to represent a space depends only on the maximal and minimal desired distances and the geometry of the space. Thus although the bulk of our results are presented in the Poincaré sphere, our discussion on precision tradeoffs is fundamental to hyperbolic space.

A representation using b bits can distinguish 2^b distinct points in a space S. Suppose we wish to capture distances up to d with error tolerance ε. Concretely, say every point in the ball $B(0, d)$ must be within distance ε of a represented point. By a sphere covering argument, this requires at least $\frac{V_S(d)}{V_S(\varepsilon)}$ points to be represented, where $V_S(r)$ is the volume of a ball of radius r in the geometry. Thus at least $b = \log \frac{V_S(d)}{V_S(\varepsilon)}$ bits are needed for the representation. Notice that $V_E(d) \sim d^n$ in Euclidean $\mathbb{R}^n$ space, so this gives the correct bit complexity of $n \log(d/\varepsilon)$. In hyperbolic space, V_H is exponential instead of polynomial in d, so $O(d)$ bits are needed in the representation (for any constant tolerance). In particular, this is independent of the model of the space.

Graph embedding lower bound Now, we derive a lower bound on the bits of precision required for embedding a graph into $\mathbb{H}_2$. Afterwards we prove a result bounding the precision for our extension of Sarkar's construction for the r-dimensional Poincaré ball $\mathbb{H}_r$. Finally, we give some details on the algorithm for this extension.

We derive the lower bound by exhibiting an explicit graph and lower bounding the precision needed to represent its nodes (for any embedding of the graph into $\mathbb{H}_2$). The explicit graph G_m we consider consists of a root node with $\deg_{\max}$ chains attached to it. Each of these chains has m nodes for a total of $1 + m(\deg_{\max})$ nodes, as shown in Figure 5.

Lemma D.1. *The bits of precision needed to embed a graph with longest path ℓ is $\Omega\left(\frac{\ell}{\varepsilon} \log(\deg_{\max})\right)$.*

Proof. We first consider the case where $m = 1$. Then G_1 is a star with $1 + \deg_{\max}$ children $a_1, a_2, \ldots, a_{\deg_{\max}}$. Without loss of generality, we can place the root a_0 at the origin 0.

Let $x_i = f(a_i)$ be the embedding into $\mathbb{H}_2$ for vertex a_i for $0 \le i \le \deg_{\max}$. We begin by showing that the distortion does not increase if we equalize the distances between the origin and each child x_i. Let us write $\ell_{\max} = \max_i d_H(0, x_i)$ and $\ell_{\min} = \min_i d_H(0, x_i)$.

What is the worst-case distortion? We must consider the maximal expansion and the maximal contraction of graph distances. Our graph distances are either 1 or 2, corresponding to edges (a_0 to a_i) and paths of length 2 (a_i to a_0 to a_j). By triangle inequality, $\frac{d_H(x_i,x_j)}{2} \le \frac{d_H(0,x_i)}{2} + \frac{d_H(0,x_j)}{2} \le \ell_{\max}$. This implies that the maximal expansion $\max_{i \ne j} d_H(f(a_i), f(a_j))/d_G(a_i, a_j)$ is $\frac{\ell_{\max}}{1} = \ell_{\max}$ occurring at a parent-child edge. Similarly, the maximal contraction

is at least $\frac{1}{\ell_{\min}}$. With this,

$$D_{\text{wc}}(f) \geq \frac{\ell_{\max}}{\ell_{\min}}.$$

Equalizing the origin-to-child distances (that is, taking $\ell_{\max} = \ell_{\min}$) reduces the distortion. Moreover, these distances are a function of the norms $\|x_i\|$, so we set $\|x_i\| = v$ for each child.

Next, observe that since there are $\deg_{\max}$ children to place, there exists a pair of children x, y so that the angle formed by $x, 0, y$ is no larger than $\theta = \frac{2\pi}{\deg_{\max}}$. In order to get a worst-case distortion of $1 + \varepsilon$, we need the product of the maximum expansion and maximum contraction to be no more than $1 + \varepsilon$. The maximum expansion is simply $d_H(0, x)$ while the maximum contraction is $\frac{2}{d_H(x,y)}$, so we wan

$$2d_H(0, x) \leq (1 + \varepsilon)d_H(x, y).$$

We use the log-based expressions for hyperbolic distance:

$$d_H(0, x) = \log\left(\frac{1 + v}{1 - v}\right),$$

and

$$\begin{aligned} d_H(x, y) &= 2\log\left(\frac{\|x - y\| + \sqrt{\|x\|^2\|y\|^2 - 2\langle x, y\rangle + 1}}{\sqrt{(1 - \|x\|^2)(1 - \|y\|^2)}}\right) \\ &= 2\log\left(\frac{\sqrt{2v^2(1 - \cos\theta)} + \sqrt{v^4 - 2v^2\cos\theta + 1}}{1 - v^2}\right). \end{aligned}$$

This leaves us with

$$\log\left(\frac{\sqrt{2v^2(1 - \cos\theta)} + \sqrt{v^4 - 2v^2\cos\theta + 1}}{1 - v^2}\right)(1 + \varepsilon) \geq \log\left(\frac{1 + v}{1 - v}\right).$$

Now, since $1 > v^2$, we have that $\sqrt{2(1 - \cos\theta)} \geq \sqrt{2v^2(1 - \cos\theta)}$. Some algebra shows that $\sqrt{3(1 - \cos\theta)} \geq \sqrt{v^4 - 2v^2\cos\theta + 1}$, so that we can upper bound the left-hand side to write

$$\log\left(\frac{(1 + \sqrt{\frac{3}{2}})\sqrt{2(1 - \cos\theta)}}{1 - v^2}\right)(1 + \varepsilon) \geq \log\left(\frac{1 + v}{1 - v}\right).$$

Next we use the small angle approximation $\cos(\theta) = 1 - \theta^2/2$ to get $\sqrt{2(1 - \cos\theta)} = \theta$. Now we have

$$\log\left(\frac{(1 + \sqrt{\frac{3}{2}})\theta}{1 - v^2}\right)(1 + \varepsilon) \geq \log\left(\frac{1 + v}{1 - v}\right).$$

Since $v < 1$, $\frac{1}{1-v} > \frac{1}{1-v^2}$ and $\frac{1+v}{1-v} \geq \frac{1}{1-v}$, so we can upper bound the left-hand side and lower bound the right-hand side:

$$\log\left(\frac{(1 + \sqrt{\frac{3}{2}})\theta}{1 - v}\right)(1 + \varepsilon) \geq \log\left(\frac{1}{1 - v}\right).$$

Rearranging,

$$-\log(1 - v) \geq -\log\left(\left(1 + \sqrt{\frac{3}{2}}\right)\theta\right)\frac{1 + \varepsilon}{\varepsilon}.$$

Recall that $\theta = \frac{2\pi}{\deg_{\max}}$. Then we have that

$$-\log(1-v) \geq \left(\frac{1+\varepsilon}{\varepsilon}\right)\left(\log(\deg_{\max}) - \log((2+\sqrt{6})\pi)\right),$$

so that

$$-\log(1-v) = \Omega\left(\frac{1}{\varepsilon}\log(\deg_{\max})\right).$$

Since $v = \|x\| = \|y\|$, $-\log(1-v)$ is precisely the required number of bits of precision, so we have our lower bound for the $m = 1$ case.

Next we analyze the $m > 1$ case. Consider the embedded vertices $x_1, x_2, \ldots, x_m$ corresponding to one chain and $y_1, y_2, \ldots, y_m$ corresponding to another. There exists a pair of chains such that the angle formed by $x_m, 0, y_1$ is at most $\theta = \frac{2\pi}{\deg_{\max}}$. Let $u = \|x_m\|$ and $v = \|y_1\|$. From the $m = 1$ case, we have a lower bound on $-\log(1-v)$; we will now lower bound $-\log(1-u)$. The worst-case distortion we consider uses the contraction given by the path $x_m \to x_{m-1} \to \cdots \to x_1 \to 0 \to y_1$; this path has length $m+1$. The expansion is just the edge between 0 and y_1. Then, to satisfy the worst-case distortion $1+\varepsilon$, we need

$$(m+1)d_H(0, y_1) \leq (1+\varepsilon)d_H(x_m, y_1).$$

Using the hyperbolic distance formulas, we can rewrite this as

$$2\log\left(\frac{\|x_m - y_1\| + \sqrt{\|x_m\|^2\|y_1\|^2 - 2\langle x_m, y_1\rangle + 1}}{\sqrt{(1-\|x_m\|^2)(1-\|y_1\|^2)}}\right)(1+\varepsilon) \geq (m+1)\log\left(\frac{1+v}{1-v}\right),$$

or,

$$2\log\left(\frac{\sqrt{u^2+v^2-2uv\cos\theta} + \sqrt{u^2v^2-2uv\cos\theta+1}}{\sqrt{(1-u^2)(1-v^2)}}\right)(1+\varepsilon) \geq (m+1)\log\left(\frac{1+v}{1-v}\right).$$

Next,

$$\begin{aligned}
&2\log\left(\frac{\sqrt{u^2+v^2-2uv\cos\theta} + \sqrt{u^2v^2-2uv\cos\theta+1}}{\sqrt{(1-u^2)(1-v^2)}}\right) \\
&\leq 2\log\left(\frac{(1+\sqrt{\tfrac{3}{2}})\theta}{\sqrt{(1-u^2)(1-v^2)}}\right) = \log\left(\frac{(1+\sqrt{\tfrac{3}{2}})^2\theta^2}{(1-u^2)(1-v^2)}\right) \\
&\leq \log\left(\frac{(1+\sqrt{\tfrac{3}{2}})^2\theta^2}{(1-u)(1-v)}\right).
\end{aligned}$$

In the first step, we used the same arguments as earlier. Applying this result and using $\frac{1+v}{1-v} \geq \frac{1}{1-v}$, we have

$$\log\left(\frac{(1+\sqrt{\tfrac{3}{2}})^2\theta^2}{(1-u)(1-v)}\right)(1+\varepsilon) \geq (m+1)\log\left(\frac{1}{1-v}\right),$$

or,

$$\log\left(\frac{(1+\sqrt{\tfrac{3}{2}})^2\theta^2}{1-u}\right)(1+\varepsilon) \geq (m-\varepsilon)\log\left(\frac{1}{1-v}\right).$$

Next we can apply the bound on $-\log(1-v)$.

$$\begin{aligned}\log\left(\frac{1}{1-u}\right) &\geq -\log\left((1+\sqrt{\frac{3}{2}})^2\theta^2\right) + \left(\frac{m-\varepsilon}{1+\varepsilon}\right)\log\left(\frac{1}{1-v}\right) \\ &\geq -\log\left((1+\sqrt{\frac{3}{2}})^2\theta^2\right) + \left(\frac{m-\varepsilon}{1+\varepsilon}\right)\left(\frac{1+\varepsilon}{\varepsilon}\right)\left(\log(\deg_{\max}) - \log((2+\sqrt{6})\pi)\right)) \\ &= \left(\frac{m-\varepsilon}{\varepsilon}\right)\log(\deg_{\max}) - \left(\frac{m-\varepsilon}{\varepsilon}\right)\log((2+\sqrt{6})\pi) - \frac{1}{2}\left(\log(\deg_{\max}) - \log((2+\sqrt{6})\pi)\right).\end{aligned}$$

Here, we applied the relationship between θ and $\deg_{\max}$ we derived earlier. To conclude, note that the longest path in our graph is $\ell = 2m$. Then, we have that

$$-\log(1-u) = \Omega\left(\frac{\ell}{\varepsilon}\log(\deg_{\max})\right),$$

as desired. □

Combinatorial construction upper bounds Next, we prove our extension of Sarkar's construction for $\mathbb{H}_r$, restated below.

Proposition 3.1. *The generalized $\mathbb{H}_r$ combinatorial construction has distortion at most $1+\varepsilon$ and requires at most $O(\frac{1}{\varepsilon}\frac{\ell}{r}\log\deg_{\max})$ bits to represent a node component for $r \leq (\log\deg_{\max})+1$, and $O(\frac{1}{\varepsilon}\ell)$ bits for $r > (\log\deg_{\max})+1$.*

Proof. The combinatorial construction achieves worst-case distortion bounded by $1+\varepsilon$ in two steps (Sarkar, 2011). First, it is necessary to scale the embedded edges by a factor of τ sufficiently large to enable each child of a parent node to be placed in a disjoint cone. Note that there will be a cone with angle α less than $\frac{\pi}{\deg_{\max}}$. The connection between this angle and the scaling factor τ is governed by $\tau = -\log(\tan\alpha/2)$. As expected, as $\deg_{\max}$ increases, α decreases, and the necessary scale τ increases.

This initial step provides a Delaunay embedding (and thus a MAP of 1.0), but perhaps not sufficient distortion. The second step is to further scale the points by a factor of $\frac{1+\varepsilon}{\varepsilon}$; this ensures the distortion upper bound.

Our generalization to the Poincaré ball of dimension r will modify the first step by showing that we can pack more children around a parent while maintaining the same angle. In other words, for a fixed number of children we can increase the angle between them, correspondingly decreasing the scale. We use the following generalization of cones for $\mathbb{H}_r$, defined by the maximum angle $\alpha \in [0, \pi/2]$ between the axis and any point in the cone. Let cone $C(X, Y, \alpha)$ be the cone at point X with axis $\vec{XY}$ and cone angle α: $C(X,Y,\alpha) = \{Z \in \mathbb{H}_r : \langle Z-X, Y-X\rangle \geq \|Z-X\|\|Y-X\|\cos\alpha\}$. We seek the maximum angle α for which $\deg_{\max}$ disjoint cones can be fit around a sphere.

Supposing $r-1 \leq \log\deg_{\max}$, we use the following lower bound (Jenssen et al., 2018) on the number of unit vectors $A(r,\theta)$ that can be placed on the unit sphere of dimension r with pairwise angle at least θ:

$$A(r,\theta) \geq (1+o(1))\sqrt{2\pi r}\frac{\cos\theta}{(\sin\theta)^{r-1}}.$$

Consider taking angle

$$\theta = \operatorname{asin}(\deg_{\max}{}^{-\frac{1}{r-1}}).$$

Note that

$$\deg_{\max}{}^{-\frac{1}{r-1}} = \exp\log\deg_{\max}{}^{-\frac{1}{r-1}} = \exp\left(-\frac{\log d}{r-1}\right) \leq 1/e,$$

which implies that θ is bounded from above and $\cos\theta$ is bounded from below. Therefore

$$\deg_{\max} = \frac{1}{(\sin\theta)^{r-1}} \leq O(1)\frac{\cos\theta}{(\sin\theta)^{r-1}} \leq A(r,\theta).$$

So it is possible to place $\deg_{\max}$ children around the sphere with pairwise angle θ, or equivalently place $\deg_{\max}$ disjoint cones with cone angle $\alpha = \theta/2$. Note the key difference compared to the two-dimensional case where $\alpha = \frac{\pi}{\deg_{\max}}$; here we reduce the angle's dependence on the degree by an exponent of $\frac{1}{r-1}$.

It remains to compute the explicit scaling factor τ that this angle yields; recall that $\tau = -\log(\tan \alpha/2)$ suffices (Sarkar, 2011). We then have

$$\begin{aligned}\tau &= -\log(\tan(\theta/4)) = -\log\left(\frac{\sin(\theta/2)}{1+\cos(\theta/2)}\right) = \log\left(\frac{1+\cos(\theta/2)}{\sin(\theta/2)}\right) \\ &\leq \log\left(\frac{2}{\sin(\theta/2)}\right) = \log\left(\frac{4\cos(\theta/2)}{\sin\theta}\right) \\ &\leq \log\left(\frac{4}{\deg_{\max}{}^{-\frac{1}{r-1}}}\right) = O\left(\frac{1}{r}\log\deg_{\max}\right).\end{aligned}$$

This quantity tells us the scaling factor without considering distortion (the first step). To yield the $1+\varepsilon$ distortion, we just increase the scaling by a factor of $\frac{1+\varepsilon}{\varepsilon}$. The longest distance in the graph is the longest path ℓ multiplied by this quantity.

Putting it all together, for a tree with longest path ℓ, maximum degree $\deg_{\max}$ and distortion at most $1+\varepsilon$, the components of the embedding require (using the fact that distances $\|d\|$ require d bits),

$$O\left(\frac{1}{\varepsilon}\frac{\ell}{r}\log d_{\max}\right)$$

bits per component. This big-O is with respect to $\deg_{\max}$ and any $r \leq \log\deg_{\max}+1$.

When $r > \log\deg_{\max}+1$, $O\left(\frac{1}{\varepsilon}\ell\right)$ is a trivial upper bound. Note that this cannot be improved asymptotically: As $\deg_{\max}$ grows, the minimum pairwise angle approaches $\pi/2$,[5] so that $\tau = \Omega(1)$ irrespective of the dimension r. □

Next, we provide more details on the coding-theoretic child placement construction for r-dimensional embeddings. Recall that children are placed at the vertices of a hypercube inscribed into the unit hypersphere, with components in $\frac{\pm 1}{\sqrt{r}}$. These points are indexed by sequences $a \in \{0,1\}^r$ so that

$$x_a = \left(\frac{(-1)^{a_1}}{\sqrt{r}}, \frac{(-1)^{a_2}}{\sqrt{r}}, \ldots, \frac{(-1)^{a_r}}{\sqrt{r}}\right).$$

The Euclidean distance between x_a and x_b is a function of the Hamming distance $d_{\text{Hamming}}(a,b)$ between a and b. The Euclidean distance is exactly $2\sqrt{\frac{d_{\text{Hamming}}(a,b)}{r}}$. Therefore, we can control the distances between the children by selecting a set of binary sequences with a prescribed minimum Hamming distance—a binary error-correcting code—and placing the children at the resulting hypercube vertices.

We introduce a small amount of terminology from coding theory. A binary code $\mathcal{C}$ is a set of sequences $\mathbf{a} \in \{0,1\}^r$. A $[r,k,h]_2$ code $\mathcal{C}$ is a binary linear code with length r (i.e., the sequences are of length r), size 2^k (there are 2^k sequences), and minimum Hamming distance h (the minimum Hamming distance between two distinct members of the code is h).

The Hadamard code $\mathcal{C}$ has parameters $[2^k, k, 2^{k-1}]$. If $r = 2^k$ is the dimension of the space, the Hamming distance between two members of $\mathcal{C}$ is at least $2^{k-1} = r/2$. Then, the distance between two distinct vertices of the hypercube x_a and x_b is $2\sqrt{\frac{r/2}{r}} = 2\sqrt{1/2} = \sqrt{2}$. Moreover, we can place up to $2^k = r$ points at least at this distance.

To build intuition, consider placing children on the unit circle ($r = 2$) compared to the $r = 128$-dimensional unit sphere. For $r = 2$, we can place up to 4 points with pairwise distance at least $\sqrt{2}$. However, for $r = 128$, we can place up to 128 children while maintaining this distance.

[5] Given points $x_1, \ldots, x_n$ on the unit sphere, $0 \leq \|\sum x_i\|_2^2 = n + \sum_{i\neq j}\langle x_i, x_j\rangle$ implies there is a pair such that $x_i \cdot x_j \geq -\frac{1}{n-1}$, i.e. an angle bounded by $\cos^{-1}(-1/(n-1))$.

We briefly describe a few more practical details. Note that the Hadamard code is parametrized by k. To place $c+1$ children, take $k = \lceil \log_2(c+1) \rceil$. However, the desired dimension r' of the embedding might be larger than the resulting code length $r = 2^k$. We can deal with this by repeating the codeword. If there are r' dimensions and $r|r'$, then the distance between the resulting vertices is still at least $\sqrt{2}$. Also, recall that when placing children, the parent node has already been placed. Therefore, we perform the placement using the hypercube, and rotate the hypersphere so that one of the $c+1$ placed nodes is located at this parent.

Embedding the ancestor transitive closure Prior work embeds the transitive closure of the WordNet noun hypernym graph (Nickel & Kiela, 2017). Here, edges are placed between each word and its hypernym ancestors; MAP is computed over edges of the form (word, hypernym), or, equivalently, edges (a, b) where $b \in \mathcal{A}(a)$ is an ancestor of a.

In this section, we show how to achieve arbitrarily good MAP on these types of transitive closures of a tree by embedding a weighted version of the tree (which we can do using the combinatorial construction with arbitrarily low distortion for any number dimensions). The weights are simply selected to ensure that nodes are always nearer to their ancestors than to any other node.

Let $T = (V, E)$ be our original graph. We recursively produce a weighted version of the graph called T' that satisfies the desired property. Let s be the depth of node $a \in V$. We weight each of the edges (a, c), where c is a child of a with weight 2^s. Now we show the following property:

Proposition D.2. *Let $b \in \mathcal{A}(a)$ be an ancestor of a and $e \notin \mathcal{A}(a)$ be some node not an ancestor of a. Then,*

$$d_G(a, b) < d_G(a, e).$$

Proof. Let a be at depth s. First, the farthest ancestor from a is the root, at distance $2^{s-1} + 2^{s-2} + \ldots + 2 + 1 = 2^s - 1$. Thus $d_G(a, b) \leq 2^s - 1$.

If e is a descendant of a, then $d_G(a, e)$ is at least 2^s Next, if e is neither a descendant nor an ancestor of a, let f be their nearest common ancestor, and let the depths of a, e, f be s, s_2, s_3, respectively, where $s_3 < \min\{s_1, s_2\}$. We have that

$$\begin{aligned} d_G(a, e) &= (2^{s-1} + \ldots + 2^{s_3}) + (2^{s_2-1} + \ldots + 2^{s_3}) \\ &= 2^s - 2^{s_3} + 2^{s_2} - 2^{s_3} \\ &= 2^s + 2^{s_2} - 2^{s_3+1} \\ &\geq 2^s \\ &> d_G(a, b). \end{aligned}$$

The fourth line follows from $s_2 > s_3$. This concludes the argument. □

Therefore, embedding the weighted tree T' with the combinatorial construction enables us to keep all of a word's ancestors nearer to it than any other word. This enables us to embed a transitive closure hierarchy (like WordNet's) while still embedding a nearly tree-like graph. [6] Furthermore, the desirable properties of the construction still carry through (perfect MAP on trees, linear-time, etc).

E. Proof of h-MDS Results

We first prove the condition that $X^T u = 0$ is equivalent to pseudo-Euclidean centering.

[6] Note that further separation can be achieved by picking weights with a base larger than 2.

Proof of Lemma 4.1. In the hyperboloid model, the variance term Ψ can be written as

$$\begin{aligned}\Psi(z; x_1, x_2, \ldots, x_n) &= \sum_{i=1}^{k} \sinh^2(d_H(x_i, z)) \\ &= \sum_{i=1}^{k} \left(\cosh^2(d_H(x_i, z)) - 1\right) \\ &= \sum_{i=1}^{k} \left((x_i^T Q z)^2 - 1\right) \\ &= \sum_{i=1}^{k} \left((x_{0,i} z_0 - \vec{x}_i^T \vec{z})^2 - 1\right) \\ &= \sum_{i=1}^{k} \left(\left(x_{0,i}\sqrt{1 + \|\vec{z}\|^2} - \vec{x}_i^T \vec{z}\right)^2 - 1\right).\end{aligned}$$

The derivative of this with respect to $\vec{z}$ is

$$\nabla_{\vec{z}} \Psi(z; x_1, x_2, \ldots, x_n) = 2 \sum_{i=1}^{k} \left(x_{0,i}\sqrt{1 + \|\vec{z}\|^2} - \vec{x}_i^T \vec{z}\right) \left(x_{0,i} \frac{\vec{z}}{\sqrt{1 + \|\vec{z}\|^2}} - \vec{x}_i\right).$$

At $\vec{z} = 0$ (or equivalently $z = e_0$), this becomes

$$\begin{aligned}\nabla_{\vec{z}} \Psi(z; x_1, x_2, \ldots, x_n)|_{\vec{z}=0} &= 2 \sum_{i=1}^{k} \left(x_{0,i}\sqrt{1 + 0} - 0\right) \left(x_{0,i} \frac{0}{\sqrt{1 + 0}} - \vec{x}_i\right) \\ &= -2 \sum_{i=1}^{k} x_{0,i} \vec{x}_i.\end{aligned}$$

If we define the matrix $X \in \mathbb{R}^{n \times k}$ such that $X^T e_i = \vec{x}_i$ and the vector $u \in \mathbb{R}^k$ such that $u_i = x_{0,i}$, then

$$\begin{aligned}\nabla_{\vec{z}} \Psi(z; x_1, x_2, \ldots, x_n)|_{\vec{z}=0} &= -2 \sum_{i=1}^{k} X^T e_i e_i^T u \\ &= -2 X^T u.\end{aligned}$$

□

Centering and Geodesic Submanifolds A well-known property of the hyperboloid model is that the geodesic submanifolds on $\mathbb{M}_r$ are exactly the linear subspaces of $\mathbb{R}^{r+1}$ intersected with the hyperboloid model (Corollary A.5.5. from (Benedetti & Petronio, 1992)). This is analogous to how the affine subspaces of $\mathbb{R}^r$ are the linear subspaces of $\mathbb{R}^{r+1}$ intersected with the homogeneous-coordinates model of $\mathbb{R}^r$. Notice that this directly implies that any geodesic submanifold can be written as a geodesic submanifold centered on any of the points in that manifold. To be explicit with the definitions:

Definition E.1. *A geodesic submanifold is a subset S of a manifold such that for any two points $x, y \in S$, the geodesic from x to y is fully contained within S.*

Definition E.2. *A geodesic submanifold rooted at a point x, given some local subspace of its tangent bundle T, is the subset S of the manifold that is the union of all the geodesics through x that are tangent at x in a direction contained in T.*

Now we prove that centering with the pseudo-Euclidean mean preserves geodesic submanifolds.

First, we need the following technical lemma showing that projection to a manifold decreases distances.

Lemma E.3. *Consider a dimension-r geodesic submanifold S and point $\bar{x}$ outside of it. Let z be the projection of $\bar{x}$ onto S. Then for any point $x \in S$, $d_H(x, \bar{x}) > d_H(x, z)$.*

Proof. As a consequence of the projection, the points $x, z, \bar{x}$ form a right angle. From the hyperbolic Pythagorean theorem, we know that

$$\cosh(d_H(x, \bar{x})) = \cosh(d_H(x, z)) \cosh(d_H(z, \bar{x})).$$

Since $\cosh$ is increasing and at least 1 (with equality only at $\cosh(0) = 1$), this implies that

$$d_H(x, \bar{x}) > d_H(x, z).$$

□

Lemma E.4. *If some points $x_1, \ldots, x_k$ lie in a dimension-r geodesic submanifold S, then both a Karcher mean and a pseudo-Euclidean mean lie in this submanifold. Equivalently, if the points lie in a submanifold, then this submanifold can be written as centered at the Karcher mean or the pseudo-Euclidean mean.*

Proof. Suppose by way of contradiction that there is a Karcher mean $\bar{x}$ that lies outside this submanifold S. Then, consider the projection z of $\bar{x}$ onto S. From Lemma E.3, projecting onto S has strictly decreased the distance to all the points on S.

As a result, the Frechet variance

$$\sum_{i=1}^{k} d_H^2(x_i, \bar{x}),$$

and decreases when $\bar{x}$ is projected onto S. From this, it follows that there is a minimum value of the Frechet variance (a Karcher mean) that lies on S. An identical argument works for the pseudo-Euclidean distance, since the pseudo-Euclidean distance uses a variance that is just the sum of monotonically increasing functions of the hyperbolic distance. □

Lemma E.5. *Given some pairwise distances $d_{i,j}$, if it is possible to embed the distances in a dimension-r geodesic submanifold rooted and centered at a pseudo-Euclidean mean, then it is possible to embed the distances in a dimension-r geodesic submanifold rooted and centered at a Karcher mean, and vice versa.*

Proof. Suppose that it is possible to embed the distances as some points $x_1, \ldots, x_k$ in a dimension-r geodesic submanifold S. Then, by Lemma E.4, S contains both a Karcher mean $\bar{x}$ and a pseudo-Euclidean mean $\bar{x}_P$ of these points. If we reflect all the points such that $\bar{x}$ is reflected to the origin, then the new reflected points will also be an embedding of the distances (since reflection is isometric) and they will also be centered at the origin. Furthermore, we know that they will still lie in a dimension-r submanifold (now containing the origin) since reflection also preserves the dimension of geodesic submanifolds. So the reflected points that we have constructed are an embedding of $d_{i,j}$ into a dimension-r geodesic submanifold rooted and centered at a Karcher mean. The same argument will show that (by reflecting $\bar{x}_P$ to the origin instead of $\bar{x}$) we can construct an embedding of $d_{i,j}$ into a dimension-r geodesic submanifold rooted and centered at the pseudo-Euclidean mean. This proves the lemma. □

F. Perturbation Analysis

F.1. Handling Perturbations

Now that we have shown that h-MDS recovers an embedding exactly, we consider the impact of perturbations on the data. Given the necessity of high precision for some embeddings, we expect that in some regimes the algorithm should be very sensitive. Our results identify the scaling of those perturbations.

First, we consider how to measure the effect of a perturbation on the resulting embedding. We measure the gap between two configurations of points, written as matrices in $\mathbb{R}^{n \times r}$, by the sum of squared differences $D(X, Y) = \text{trace}((X - Y)^T (X - Y))$. Of course, this is not immediately useful, since X and Y can be rotated or reflected without affecting the distance matrix used for MDS—as these are isometries, while scalings and Euclidean translations are not. Instead, we measure the gap by

$$D_E(X, Y) = \inf\{D(X, PY) : P^T P = I\}.$$

In other words, we look for the configuration of Y with the smallest gap relative to X. For Euclidean MDS, Sibson (1978) provides an explicit formula for $D_E(X, Y)$ and uses this formulation to build a perturbation analysis for the case where Y is a configuration recovered by performing MDS on the perturbed matrix $XX^T + \Delta(E)$, with $\Delta(E)$ symmetric.

Problem setup In our case, the perturbations affect the hyperbolic distances. Let $H \in \mathbb{R}^{n\times n}$ be the distance matrix for a set of points in hyperbolic space. Let $\Delta(H) \in \mathbb{R}^{n\times n}$ be the perturbation, with $H_{i,i} = 0$ and $\Delta(H)$ symmetric (so that $\hat{H} = H + \Delta_H$ remains symmetric). The goal of our analysis is to estimate the gap $D_E(X, Y)$ between X recovered from H with h-MDS and $\hat{X}$ recovered from the perturbed distances $H + \Delta(H)$.

Lemma F.1. *Under the above conditions, if $\lambda_{\min}$ denotes the smallest nonzero eigenvalue of XX^T then up to second order in $\Delta(H)$,*

$$D_E(X, \hat{X}) \leq \frac{2n^2}{\lambda_{\min}} \sinh^2\left(\|H\|_\infty\right) \|\Delta(H)\|_\infty^2.$$

The key takeaway is that this upper bound matches our intuition for the scaling: if all points are close to one another, then $\|H\|_\infty$ is small and the space is approximately flat (since $\sinh^2(z)$ is dominated by $2z^2$ close to the origin). On the other hand, points at great distance are sensitive to perturbations in an absolute sense.

Proof of Lemma F.1. Similarly to our development of h-MDS, we proceed by accessing the underlying Euclidean distance matrix, and then apply the perturbation analysis from Sibson (1979). There are three steps: first, we get rid of the acosh in the distances to leave us with scaled Euclidean distances. Next, we remove the scaling factors, and apply Sibson's result. Finally, we bound the gap when projecting to the Poincaré sphere.

Hyperbolic to scaled Euclidean distortion Let Y denote the scaled-Euclidean distance matrix, as in (1), so that $Y_{i,j} = \cosh(H_{i,j})$. Let $\hat{Y}_{i,j} = \cosh(H_{i,j} + \Delta(H)_{i,j})$. We write $\Delta(Y) = \hat{Y} - Y$ for the scaled Euclidean version of the perturbation. We can use the hyperbolic-cosine difference formula on each term to write

$$\begin{aligned}
\Delta(Y)_{i,j} &= \cosh(\hat{H}_{i,j}) - \cosh(H_{i,j}) \\
&= (\cosh(H_{i,j} + \Delta(H)_{i,j}) - \cosh(H_{i,j})) \\
&= 2\sinh\left(\frac{2H_{i,j} + \Delta(H)_{i,j}}{2}\right)\sinh\left(\frac{\Delta(H)_{i,j}}{2}\right).
\end{aligned}$$

In terms of the infinity norm, as long as $\|H\|_\infty \geq \|\Delta(H)\|_\infty$ (it is fine to assume this because we are only deriving a bound up to second order, so we can suppose that $\Delta(H)$ is small), we can simplify this to

$$\|\Delta(Y)\|_\infty \leq 2\sinh\left(\|H\|_\infty\right)\sinh\left(\|\Delta(H)\|_\infty/2\right).$$

Scaled Euclidean to Euclidean inner product. Recall that if X is the embedding in the hyperboloid model, then $Y = uu^T - XX^T$ and furthermore $X^T u = 0$ so that X can be recovered through PCA. Now we are in the Euclidean setting, and can thus measure the result of the perturbation on the recovered X. The proof of Theorem 4.1 in Sibson (1979) transfers to this setting. This result states that if $\hat{X}$ is the configuration recovered from the perturbed inner products, then the lowest-order term of the expansion of the error $D_E(X, \hat{X})$ in the perturbation $\Delta(Y)$ is

$$D_E(X, \hat{X}) = \frac{1}{2}\sum_{j,k} \frac{(v_j^T \Delta(Y) v_k)^2}{\lambda_j + \lambda_k}.$$

Here, the λ_i and v_i are the eigenvalues and corresponding orthonormal eigenvectors of XX^T and the sum is taken over pairs of λ_j, λ_k that are not both 0. Let $\lambda_{\min}$ be the smallest nonzero eigenvalue of XX^T. Then,

$$\begin{aligned}
D_E(X, \hat{X}) &\leq \frac{1}{2\lambda_{\min}}\sum_{j,k}(v_j^T \Delta(Y) v_k)^2 \leq \frac{1}{2\lambda_{\min}}\|\Delta(Y)\|_F^2 \\
&\leq \frac{n^2}{2\lambda_{\min}}\|\Delta(Y)\|_\infty^2.
\end{aligned}$$

Combining this with the previous bounds and restricting to second-order terms in $\|\Delta(H)\|_\infty^2$ proves Lemma F.1 for the embedding X in the hyperboloid model. □

Projecting to the Poincaré disk Algorithm 2 initially finds an embedding in $\mathbb{M}_r$, but optionally converts it to the Poincaré disk. To convert a point x in the hyperboloid model to z in the Poincaré disk, take $z = \frac{x}{1+\sqrt{1+\|x\|_2^2}}$. Let $Z \in \mathbb{R}^{n\times r}$ be the projected embedding. Now we show that the same perturbation bound holds after projection.

Lemma F.2. *For any x and y,* $\left\| \frac{x}{1+\sqrt{1+\|x\|_2^2}} - \frac{x}{1+\sqrt{1+\|x\|_2^2}} \right\| \leq \|x - y\|$

Proof. Let $u_x = \sqrt{1+\|x\|^2}$ and define u_y analogously. Note that $u_x \geq 2$, $u_x \geq \|x\|$, and

$$u_y - u_x = \frac{u_y^2 - u_x^2}{u_y + u_x} = (\|y\| - \|x\|)\frac{\|y\| + \|x\|}{u_y + u_x} \leq \|y\| - \|x\|.$$

Combining these facts leads to the bound

$$\begin{aligned}
\left\| \frac{x}{1+\sqrt{1+\|x\|_2^2}} - \frac{y}{1+\sqrt{1+\|y\|_2^2}} \right\| &= \left\| \frac{x - y + xu_y - yu_y + yu_y - yu_x}{(1+u_x)(1+u_y)} \right\| \\
&= \left\| \frac{(x-y)(1+u_y) + y(u_y - u_x)}{(1+u_x)(1+u_y)} \right\| \\
&= \left\| \frac{x-y}{1+u_x} + \frac{y}{1+u_y}\frac{u_y - u_x}{1+u_x} \right\| \\
&\leq \frac{\|x-y\|}{1+u_x} + \frac{\|u_y - u_x\|}{1+u_x} \\
&\leq \|x - y\|.
\end{aligned}$$

□

Lemma F.2 is equivalent to the statement that $D(z, \hat{z}) \leq D(x, \hat{x})$ where $z, \hat{z}$ are the projections of $x, \hat{x}$. Since orthogonal matrices P preserve the ℓ_2 norm, $P\hat{z}$ is the projection of $P\hat{x}$ so $D(z, P\hat{z}) \leq D(x, P\hat{x})$ for any P. Finally, $D(Z, P\hat{Z})$ is just a sum over all columns and therefore $D(Z, P\hat{Z}) \leq D(X, P\hat{X})$. This implies that $D_E(Z, \hat{Z}) \leq D_E(X, \hat{X})$ as desired.

The hyperbolic gap The gap $D(X, \hat{X})$ can be written as a sum $\sum d_E(x_i, \hat{x}_i)^2$ over the vectors (columns) of $X, \hat{X}$. We can instead ask about the hyperbolic gap

$$D_H(X, \hat{X}) = \inf\left\{ \sum d_H(x_i, P\hat{x}_i)^2 : P^T P = I \right\},$$

which is a better interpretation of the perturbation error when recovering hyperbolic distances.

Note that for any points x, y in the Gans model, we have

$$d_H(x,y) = \operatorname{acosh}\left(\sqrt{1+\|x\|^2}\sqrt{1+\|y\|^2} - \langle x, y\rangle\right) \leq \operatorname{acosh}\left(\frac{2+\|x\|^2+\|y\|^2}{2} - \langle x, y\rangle\right) = \operatorname{acosh}\left(1 + \frac{1}{2}\|x-y\|^2\right).$$

Furthermore, the function $\operatorname{acosh}(1+t^2/2) - t$ is always negative except in a tiny region around $t = 0$ (and attains a maximum here on the order of 10^{-10}), so effectively $\operatorname{acosh}\left(1 + \frac{1}{2}\|x-y\|^2\right) \leq \|x - y\| = d_E(x, y)$, and the same bound in Lemma F.1 carries over to the hyperbolic gap.

G. Proof of Lemma 4.3

In this section, we prove Lemma 4.3, which gives a setting under which we can guarantee that the hyperbolic PGA objective is locally convex.

Proof of Lemma 4.3. We begin by considering the component function

$$f_i(\gamma) = \operatorname{acosh}^2(1 + d_E^2(\gamma, v_i)).$$

Here, the γ is a geodesic through the origin. We can identify this geodesic on the Poincaré disk with a unit vector u such that $\gamma(t) = (2t-1)u$. In this case, simple Euclidean projection gives us

$$d_E^2(\gamma, v_i) = \|(I - uu^T)v_i\|^2.$$

Optimizing over γ is equivalent to optimizing over u, and so

$$f_i(u) = \operatorname{acosh}^2\left(1 + \|(I - uu^T)v_i\|^2\right).$$

If we define the functions

$$h(\gamma) = \operatorname{acosh}^2(1+\gamma)$$

and

$$R(u) = \|(I - uu^T)v_i\|^2 = \|v_i\|^2 - (u^T v_i)^2$$

then we can rewrite f_i as

$$f_i(u) = h(R(u)).$$

Now, optimizing over u is an geodesic optimization problem on the hypersphere. Every goedesic on the hypersphere can be isometrically parameterized in terms of an angle θ as

$$u(\theta) = x\cos(\theta) + y\sin(\theta)$$

for orthogonal unit vectors x and y. Without loss of generality, suppose that $y^T v_i = 0$ (we can always choose such a y because there will always be some point on the geodesic that is orthogonal to v_i). Then, we can write

$$R(\theta) = \|v_i\|^2 - (x^T v_i)^2\cos^2(\theta) = \|v_i\|^2 - (x^T v_i)^2 + (x^T v_i)^2\sin^2(\theta).$$

Differentiating the objective with respect to θ,

$$\begin{aligned}\frac{d}{d\theta}h(R(\theta)) &= h'(R(\theta))R'(\theta)\\ &= 2h'(R(\theta))\cdot(v_i^T x)^2\cdot\sin(\theta)\cos(\theta).\end{aligned}$$

Differentiating again,

$$\frac{d^2}{d\theta^2}h(R(\theta)) = 4h''(R(\theta))\cdot(v_i^T x)^4\cdot\sin^2(\theta)\cos^2(\theta) + 2h'(R(\theta))\cdot(v_i^T x)^2\cdot\left(\cos^2(\theta) - \sin^2(\theta)\right).$$

Now, suppose that we are interested in the Hessian at a point $z = x\cos(\theta) + y\sin(\theta)$ for some fixed angle θ. Here, $R(\theta) = R(z)$, and as always $v_i^T z = v_i^T x\cos(\theta)$, so

$$\begin{aligned}\frac{d^2}{d\theta^2}h(R(\theta))\big|_{u(\theta)=z} &= 4h''(R(\theta))\cdot(v_i^T x)^4\cdot\sin^2(\theta)\cos^2(\theta) + 2h'(R(\theta))\cdot(v_i^T x)^2\cdot\left(\cos^2(\theta) - \sin^2(\theta)\right)\\ &= 4h''(R(z))\cdot\frac{(v_i^T z)^4}{\cos^4(\theta)}\cdot\sin^2(\theta)\cos^2(\theta) + 2h'(R(z))\cdot\frac{(v_i^T x)^2}{\cos^2(\theta)}\cdot\left(\cos^2(\theta) - \sin^2(\theta)\right)\\ &= 4h''(R(z))\cdot(v_i^T z)^4\cdot\tan^2(\theta) + 2h'(R(z))\cdot(v_i^T z)^2\cdot\left(1 - \tan^2(\theta)\right)\\ &= 2h'(R(z))\cdot(v_i^T z)^2 + \left(4h''(R(z))\cdot(v_i^T z)^4 - 2h'(R(z))\cdot(v_i^T z)^2\right)\tan^2(\theta).\end{aligned}$$

But we know that since h is concave and increasing, this last expression in parenthesis must be negative. It follows that a lower bound on this expression for fixed z will be attained when $\tan^2(\theta)$ is maximized. For any geodesic through z, the angle θ is the distance along the geodesic to the point that is (angularly) closest to v_i. By the Triangle inequality, this will be no greater than the distance θ along the Geodesic that connects z with the normalization of v_i. On this worst-case geodesic,

$$v_i^T z = \|v_i\|\cos(\theta),$$

and so

$$\cos^2(\theta) = \frac{(v_i^T z)^2}{\|v_i\|^2}$$

and

$$\tan^2(\theta) = \sec^2(\theta) - 1 = \frac{\|v_i\|^2}{(v_i^T z)^2} - 1 = \frac{R(z)}{(v_i^T z)^2}.$$

Thus, for any geodesic, for the worst-case angle θ,

$$\begin{aligned}\frac{d^2}{d\theta^2} h(R(\theta))\big|_{u(\theta)=z} &\geq 2h'(R(z)) \cdot (v_i^T z)^2 + \left(4h''(R(z)) \cdot (v_i^T z)^4 - 2h'(R(z)) \cdot (v_i^T z)^2\right) \tan^2(\theta) \\ &= 2h'(R(z)) \cdot (v_i^T z)^2 + \left(4h''(R(z)) \cdot (v_i^T z)^2 - 2h'(R(z))\right) R(z).\end{aligned}$$

From here, it is clear that this lower bound on the second derivative (and as a consequence local convexity) is a function solely of the norm of v_i and the residual to z. From simple evaluation, we can compute that

$$h'(\gamma) = 2\frac{\operatorname{acosh}(1+\gamma)}{\sqrt{\gamma^2 + 2\gamma}}$$

and

$$h''(x) = 2\frac{\sqrt{\gamma^2 + 2\gamma} - (1+\gamma)\operatorname{acosh}(1+\gamma)}{(\gamma^2 + 2\gamma)^{3/2}}.$$

As a result

$$\begin{aligned}4\gamma h''(\gamma) + h'(\gamma) &= 8\frac{\gamma\sqrt{\gamma^2 + 2\gamma} - (\gamma^2 + \gamma)\operatorname{acosh}(1+\gamma)}{(\gamma^2 + 2\gamma)^{3/2}} + 2\frac{(\gamma^2 + 2\gamma)\operatorname{acosh}(1+\gamma)}{(\gamma^2 + 2\gamma)^{3/2}} \\ &= 2\frac{4\gamma\sqrt{\gamma^2 + 2\gamma} - 4(\gamma^2 + \gamma)\operatorname{acosh}(1+\gamma) + (\gamma^2 + 2\gamma)\operatorname{acosh}(1+\gamma)}{(\gamma^2 + 2\gamma)^{3/2}} \\ &= 2\frac{4\gamma\sqrt{\gamma^2 + 2\gamma} - (3\gamma^2 + 2\gamma)\operatorname{acosh}(1+\gamma)}{(\gamma^2 + 2\gamma)^{3/2}}.\end{aligned}$$

For any γ that satisfies $0 \leq \gamma \leq 1$,

$$4\gamma\sqrt{\gamma^2 + 2\gamma} \geq (3\gamma^2 + 2\gamma)\operatorname{acosh}(1+\gamma)$$

and so

$$4\gamma h''(\gamma) + h'(\gamma) \geq 0.$$

Thus, if $0 \leq R(z) \leq 1$,

$$\begin{aligned}\frac{d^2}{d\theta^2} h(R(\theta))\big|_{u(\theta)=z} &\geq 2h'(R(z)) \cdot (v_i^T z)^2 + \left(4h''(R(z)) \cdot (v_i^T z)^2 - 2h'(R(z))\right) R(z) \\ &= h'(R(z)) \cdot (v_i^T z)^2 + (4h''(R(z)) \cdot R(z) + h'(R(z))) \cdot (v_i^T z)^2 - 2h'(R(z)) \cdot R(z) \\ &\geq h'(R(z)) \cdot (v_i^T z)^2 - 2h'(R(z)) \cdot R(z) \\ &= h'(R(z)) \cdot \left(\|v_i\|^2 - R(z)\right) - 2h'(R(z)) \cdot R(z) \\ &= h'(R(z)) \cdot \left(\|v_i\|^2 - 3R(z)\right).\end{aligned}$$

Thus, a sufficient condition for convexity is for (as we assumed above) $R(z) \leq 1$ and

$$\|v_i\|^2 \geq 3R(z).$$

Combining these together shows that if

$$\operatorname{acosh}^2\left(1 + d_E(\gamma, v_i)^2\right) = R(z) \leq \min\left(1, \frac{1}{3}\|v_i\|^2\right)$$

then f_i is locally convex at z. The result of the lemma now follows from the fact that f is the sum of many f_i and the sum of convex functions is also convex. □

Dataset	Nodes	Edges	C-$\mathbb{H}_2$ MAP
Wikipedia Category Hierarchy	77836	151941	0.604

Table 6. MAP measure for Wikipedia category hierarchy. Closer to 1 is better.

					$\varepsilon = 0.1$		$\varepsilon = 1.0$	
Dataset	Nodes	Edges	$d_{\max}$	Time	Scaling Factor	Precision	Scaling Factor	Precision
Bal. Tree 1	40	39	4	3.78	23.76	102	4.32	18
Phylo. Tree	344	343	16	3.13	55.02	2361	10.00	412
WordNet	74374	75834	404	1346.62	126.11	2877	22.92	495
CS PhDs	1025	1043	46	4.99	78.30	2358	14.2	342
Diseases	516	1188	24	3.92	63.97	919	13.67	247
Protein - Yeast	1458	1948	54	6.23	81.83	1413	15.02	273
Gr-QC	4158	13428	68	75.41	86.90	1249	16.14	269
California	5925	15770	105	114.41	96.46	1386	19.22	245

Table 7. Combinatorial construction parameters and results.

H. Experimental Results

In this section, we provide some additional experimental results and details. We also present results on two additional less tree-like graphs (a search engine query response graph for the search term 'California' (Kleinberg, 1999) and a page category hierarchy for Wikipedia built using the WikiPrep tool (Gabrilovich & Markovitch, 2007)). For the latter graph, we used the combinatorial construction with parameter $\varepsilon = 0.1$; the result is shown in Table 6.

Combinatorial Construction: Parameters To improve the intuition behind the combinatorial construction, we report some additional parameters used by the construction. For each of the graphs, we report the maximum degree, the scaling factor τ that the construction used (note how these vary with the size of the graph and the maximal degree), the time it took to perform the embedding in seconds, and the number of bits needed to store a component for $\varepsilon = 0.1$ and $\varepsilon = 1.0$.

WordNet Relationship Embedding Details Table 4 is based off previous work on compressing entity-relationship-entity triples, and measures how effectively the embedding preserves relationship knowledge. Three relationship graphs were embedded separately, and a relationship prediction for a pair of entities is given by the embedding in which they are closest. We test pairs of entities based off Socher et al. (2013)'s train set, where correctness means the top prediction coincides with the true relationship (the graph where they are 1 edge apart).

Hyperparameter: Effect of Rank We also considered the influence of the dimension on the performance of h-MDS, PCA, and FB. On the Phylogenetic tree dataset, we measured distortion and MAP metrics for dimensions of 2,5,10,50,100, and 200. The results are shown in Table 8. We expected all of the techniques to improve with larger rank, and this was the case as well. Here, the optimization-based approach typically produces the best MAP, optimizing the fine details accurately. We observe that the gap is closed when considering 2-MAP (that is, MAP where the retrieved neighbors are at distance up

	MAP			2-MAP			d_{avg}		
Rank	h-MDS	PCA	FB	h-MDS	PCA	FB	h-MDS	PCA	FB
Rank 2	0.346	0.614	**0.718**	0.754	**0.874**	0.802	**0.317**	0.888	0.575
Rank 5	0.439	0.627	**0.761**	0.844	0.905	**0.950**	**0.083**	0.833	0.583
Rank 10	0.471	0.632	**0.777**	0.857	0.912	**0.953**	**0.048**	0.804	0.586
Rank 50	0.560	0.687	**0.784**	0.880	0.962	**0.974**	**0.036**	0.768	0.584
Rank 100	0.645	0.698	**0.795**	0.926	**0.999**	0.981	**0.036**	0.760	0.583
Rank 200	0.823	**1.0**	0.811	0.968	**1.0**	0.986	**0.039**	0.746	0.583

Table 8. Phylogenetic tree dataset. Variation with rank, measured with MAP, 2-MAP, and d_{avg}.

Rank	No Scale	Learned Scale	Exp. Weighting
50	0.481	0.508	**0.775**
100	0.688	0.681	**0.882**
200	0.894	0.907	**0.963**

Table 9. Ph.D. dataset. Improved MAP performance of PyTorch implementation using a modified PGA-like loss function.

Precision	D_{avg}	MAP
128	0.357	0.347
256	0.091	0.986
512	0.076	1.0
1024	0.064	1.0

Table 10. h-MDS recovery at different precision levels for a 3-ary tree and rank 10.

to 2 away). In particular we see that the main limitation of h-MDS is at the finest layer, confirming the idea that MAP is heavily influenced by local changes. In terms of distortion, we found that h-MDS offers good performance even at a very low dimension (0.083 at 5 dimensions).

Learned Scale In Table 9, we verify the importance of scaling that our analysis suggests; our PyTorch implementation has a simple learned scale parameter. Moreover, we added an exponential weighting to the distances in order to penalize long paths, thus improving the local reconstruction. These techniques indeed improve the MAP; in particular, the learned scale provides a better MAP at lower rank. We hope these techniques can be useful in other embedding techniques.

Precision Experiment (cf Table 10). Finally, we considered the effect of precision on h-MDS for a balanced tree and fixed dimension 10.