This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Nested Hyperbolic Spaces for Dimensionality Reduction and Hyperbolic NN Design

Xiran Fan
University of Florida
Department of Statistics
fanxiran@ufl.edu

Chun-Hao Yang
National Taiwan University
Institute of Applied Mathematical Sciences
chunhaoy@ntu.edu.tw

Baba C. Vemuri
University of Florida
Department of CISE
vemuri@ufl.edu

Abstract

Hyperbolic neural networks have been popular in the recent past due to their ability to represent hierarchical data sets effectively and efficiently. The challenge in developing these networks lies in the nonlinearity of the embedding space namely, the Hyperbolic space. Hyperbolic space is a homogeneous Riemannian manifold of the Lorentz group which is a semi-Riemannian manifold, i.e. a manifold equipped with an indefinite metric. Most existing methods (with some exceptions) use local linearization to define a variety of operations paralleling those used in traditional deep neural networks in Euclidean spaces. In this paper, we present a novel fully hyperbolic neural network which uses the concept of projections (embeddings) followed by an intrinsic aggregation and a nonlinearity all within the hyperbolic space. The novelty here lies in the projection which is designed to project data on to a lower-dimensional embedded hyperbolic space and hence leads to a nested hyperbolic space representation independently useful for dimensionality reduction. The main theoretical contribution is that the proposed embedding is proved to be isometric and equivariant under the Lorentz transformations, which are the natural isometric transformations in hyperbolic spaces. This projection is computationally efficient since it can be expressed by simple linear operations, and, due to the aforementioned equivariance property, it allows for weight sharing. The nested hyperbolic space representation is the core component of our network and therefore, we first compare this representation – independent of the network – with other dimensionality reduction methods such as tangent PCA, principal geodesic analysis (PGA) and HoroPCA. Based on this equivariant embedding, we develop a novel fully hyperbolic graph convolutional neural network architecture to learn the parameters of the projection. Finally, we present experiments demonstrating comparative performance of our network on several publicly available data sets.

1. Introduction

Hyperbolic geometry is a centuries old field of non-Euclidean geometry and has recently found its way into the field of machine learning, in particular into deep learning in the form of hyperbolic neural networks (HNNs) or hyperbolic graph convolutional networks (HGCNs) and recently for dimensionality reduction of data embedded in the hyperbolic space. In this paper, we will discuss both problems namely, dimensionality reduction in hyperbolic spaces and HNN architectures. In particular, we will present novel techniques for both these problems. In the following, we present literature review of the two above stated problems and establish the motivation for our work. A word on terminology, we will use the term hyperbolic neural network and hyperbolic graph (convolutional) neural network synonymously in the rest of the paper.

1.1. Dimensionality Reduction of Manifold-valued Data

Dimensionality reduction is a fundamental problem in machine learning with applications in computer vision and many other fields of engineering and sciences. The simplest and most popular method among these is the principal component analysis (PCA), which was proposed more than a century ago (see [22] for a review and some recent developments on PCA). PCA however is limited to data in vector spaces. For data that are manifold-valued, principal geodesic analysis (PGA) was presented in [10], which yields the projection of data onto principal geodesic submanifolds passing through an intrinsic (Fréchet) mean (FM) [11] of the data (assuming it exists). They find the geodesic submanifold of a lower dimension that maximizes the projected variance and computationally, this was achieved via linear approximation, i.e., applying PCA on the tangent space anchored at the FM. This is sometimes referred to as the tangent PCA (tPCA). This approximation however requires the data to be clustered around the FM, otherwise the tangent space approximation to the manifold leads to

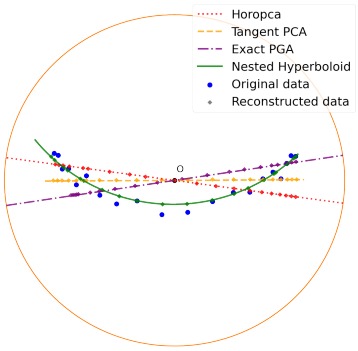


Figure 1. Projections of data from a 2D to 1D hyperbolic space using different dimensionality reduction methods. The results are visualized in the Poincaré disk. Original data (blue dots) lie in a 2D hyperbolic space and have a zero mean (origin of the Poincaré disk). The HoroPCA direction (red dotted line) and the principal geodesic obtained by tPCA (orange dashed line) and EPGA (purple dash-dotted line) fail to capture the main trend of the data since they are restricted to learn a geodesic submanifold passing through the FM. In contrast, our NH model (green solid line), captures the data trend more accurately. The diamond markers on each line represent the reconstructed data from each method. The reconstruction errors for HoroPCA, tPCA, EPGA and the NH model in this example are, 0.1708, 0.1202, 0.1638 and 0.0062 respectively.

inaccuracies. Subsequently, [42] presented the Exact PGA (EPGA) algorithm, which does not use any linear approximation. However, EPGA is computationally expensive as it requires two non-linear optimizations steps per iteration (projection to the geodesic submanifold and finding the new geodesic direction such that the reconstruction error is minimized). Later, authors in [4] developed a version of EPGA for constant sectional curvature manifolds, namely the hypersphere and the hyperbolic space, by deriving closed form formulae for the projection. There are many variants of PGA and we refer the reader to [1, 21, 48] for the details. More recently, Barycentric subspace analysis (BSA) was proposed in [36] which finds a more general parameterization of a nested sequence of submanifolds via the minimization of unexplained variance. Another useful dimensionality reduction scheme is the Principal curves [18] and their generalization to Riemannian manifolds [19] that are more appropriate for certain applications.

A salient feature of PCA is that it yields nested linear subspaces, i.e., the reduced dimensional principal subspaces form a nested hierarchy. This idea was exploited in [23] where authors proposed the *principal nested spheres* (PNS) by embedding an $(n-1)$-sphere in to an n-sphere, the embedding however is not necessarily isometric. Hence, PNS is more general than PGA in that PNS does not have to be geodesic. Similarly, for the manifold P_n of $(n \times n)$ symmetric positive definite (SPD) matrices, authors in [17] proposed a geometry-aware dimensionality reduction by projecting data on P_n to P_m for some $m \ll n$. More recently, the idea of constructing a nested sequence of manifolds was presented in [47] where authors unified and generalized the nesting concept to general Riemannian homogeneous manifolds, which form a large class of Riemannian manifolds, including the hypersphere, P_n, the Grassmannian, Stiefel manifold, Lie groups, and others. Although the general framework in [47] seems straightforward and applicable to hyperbolic spaces, many significantly important technical aspects need to be addressed and derived in detail. In this paper, we will present novel derivations suited for the hyperbolic spaces – a projection operator which is proved to yield an isometric embedding, and a proof of equivariance to isometries of the projection operator – which will facilitate the construction of nested hyperbolic spaces and the hyperbolic neural network. Note that there are five models of the hyperbolic space namely, the hyperboloid (Lorentz) model, the Poincaré disk/ball model, the Poincaré half plane model, the Klein model and the Jemisphere model [3]. All these models are isometrically equivalent but some are better suited than others depending on the application. We choose the Lorentz model of the hyperbolic space with a Lorentzian metric in our work. The choice of this model and the associated metric over other models is motivated by the properties of Riemannian optimization efficiency and numerical stability afforded [7, 34].

Most recently, an elegant approach called *HoroPCA* was proposed in [5], for dimensionality reduction in hyperbolic spaces. In particular, the authors represented the hyperbolic space using the Poincaré model and they proposed to generalize the notion of direction and the coordinates in a given direction using *ideal points* (points at infinity) and the *Busemann coordinates* (defined using the *Busemann function*) [2]. The levels sets of the Busemann function, called the *horospheres*, resemble the hyperplanes (or affine subspaces) in Euclidean spaces and hence the dimensionality reduction is achieved by a projection that moves points along a horosphere. The data is then projected to a *geodesic hull of a base point* b *and a number of ideal points* $p_1, \ldots, p_K$, *which is also a geodesic submanifold. This is the key difference between HoroPCA and our proposed method which leads to a significant difference in performance.* This is evident from the toy example in Figure 1 which depicts the reduced dimensional representations obtained by our method in comparison to those from EPGA, HoroPCA, and tPCA. Note that all of the other methods yield submanifold representations that do not capture the data trend accurately, unlike ours. More comprehensive comparisons will be made in a later section.

To briefly summarize, our first goal in this paper is to present a nested hyperbolic space representation for dimensionality reduction and then demonstrate via synthesized

and real datasets, that it achieves a lower reconstruction error in comparison to competing methods.

1.2. Hyperbolic Neural Networks

Several researchers have demonstrated that the hyperbolic space is apt for modeling hierarchically organized data, for example, graphs and trees [33, 38, 39]. Recently, the formalism of Gyrovector spaces (an algebraic structure) [44] was applied to the hyperbolic space to define basic operations paralleling those in vector spaces and were used to build a hyperbolic neural network (HNN) [13, 41]. The Gyrovector space formalism facilitates performing Möbius additions and subtractions in the Poincaré model of the hyperbolic space. HNNs have been successfully applied to word embeddings [43] as well as image embeddings [24]. Additionally, several existing deep network architectures have been modified to suit hyperbolic embeddings of data, e.g., graph networks [6, 29], attention module [15], and variational auto-encoders [30, 35]. These hyperbolic networks were shown to perform comparably or even better than their euclidean counterparts.

Existing HNNs have achieved moderate to great successes in multiple areas and shown great potential in solving complex problems. However, most of them use tangent space approximations to facilitate the use of vector space operations prevalent in existing neural network architectures. There are however some exceptions, for instance, the authors in [8] developed what they call a Hyperbolic-to-Hyperbolic network and the authors in [7] also developed a fully Hyperbolic network. They both considered the use of Lorentz transformations on hyperbolic features since the Lorentz transformation matrix acts transitively on a hyperbolic space and thus preserves the global hyperbolic structure. Each Lorentz transformation is a composition of a Lorentz rotation and a rotation free Lorentz transformation called the Lorentz boost operation. Authors in [8] only use Lorentz rotation for hyperbolic feature transformations while authors in [7] build a fully-connected layer in hyperbolic space (called a hyperbolic linear layer) parameterized by an arbitrary weight matrix (not necessarily invertible) which is applied to each data point in the hyperbolic space resulting in a mapping from a hyperbolic space to itself. This procedure is ad hoc in the sense that it does not use the intrinsic characterization of the hyperbolic space as a homogeneous space with the isometry group being the Lorentz group.

Lorentz transformations are however inappropriate for defining projection operations (required for reducing the dimensionality) as they preserve the Lorentz model only *when there is no change in dimension*. In other words, to find a lower-dimensional hyperbolic space representation for data embedded in a higher-dimensional hyperbolic space, one cannot use Lorentz transformations directly. Hence, we propose to use an isometric embedding operation mentioned in the previous subsection as the building block to design a hyperbolic neural network. We will now briefly summarize our proposed model and the contributions of our work.

1.3. Proposed Model and Contributions

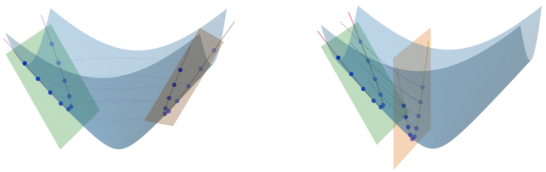
Inspired by [23] and [47], we construct a nested representation in a hyperbolic space to extract the hyperbolic features. Such a nested (hierarchical) hyperbolic space representation has the advantage that the data in reduced dimensions remains in a hyperbolic space. Hereafter, we refer to these nested hyperbolic spaces as nested hyperboloids (NHs). As a dimensionality reduction method in Riemannian manifolds, the learned lower dimensional submanifold in NH is not required to pass the FM unlike in PGA and need not be a geodesic submanifold as in HoroPCA, PGA or EPGA. In the experiments section, we will demonstrate that this leads to much lower reconstruction error in comparison to the aforementioned dimensionality reduction methods.

After defining the projection which leads to an embedding within hyperbolic spaces of different dimensions, these projections/embeddings are used to define a feature transformation layer in the hyperbolic space. This layer is then composed with a hyperbolic neighborhood aggregation operation/layer and an appropriate non-linear operations in between namely, the tangent-ReLU, to define a novel nested hyperbolic graph convolutional network (NHGCN) architecture.

The rest of the paper is organized as follows. In Section 2, we briefly review the geometry of hyperbolic space. In Section 3, we explicitly give the projection and embedding to map data between hyperbolic spaces of different dimensions. We also present a novel hyperbolic graph convolutional neural network architecture based on these projections and the tangent-ReLU activation. In Section 4, we first present the performance of the NH model as a dimensionality reduction method and compare with other competing methods, including EPGA, tPCA and HoroPCA. Next, we compare our NHGCN with other hyperbolic networks in tackling the problems of link prediction and node classification on four graph datasets described and used in [6]. Finally, we draw conclusions in Section 5.

2. Preliminaries

In this section, we briefly review relevant concepts of hyperbolic geometry. In this paper, we will regard the hyperbolic space as a homogeneous Riemannian manifold of the Lorentz group and present a few important geometric concepts, including the geodesic distance and the exponential map, in the hyperbolic space, which are used in our work. The materials presented in this section can be found in most textbooks on hyperbolic spaces, for example [3, 37].



(a) Lorentz rotation (b) Lorentz boost

Figure 2. Illustration of two kinds of Lorentz transformation, Lorentz rotation and Lorentz boost in a Lorentz model. They are isometric operations of the Lorentz model.

2.1. Lorentzian Space and Hyperbolic Space

As mentioned in Section 1, there are several (isometrically) equivalent models of a hyperbolic space, including the Poincaré model, Klein model, the upper-half space model, and the Jemisphere model [3]. We choose to use the hyperboloid (Lorentz) model of the hyperbolic space in this paper due to its numerical stability property which is very useful for the optimization problem involved in the training and test phases. Our technique is however applicable to all of the models due to the isometric equivalence of the models.

The $(n+1)$-dimensional *Lorentzian space* $\mathbb{R}^{1,n}$ is the Euclidean space $\mathbb{R}^{n+1}$ equipped with a bilinear form

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_L = -x_0 y_0 + x_1 y_1 + \cdots + x_n y_n \tag{1}$$

where $\boldsymbol{x} = [x_0, x_1, \ldots, x_n]^T, \boldsymbol{y} = [y_0, y_1, \ldots, y_n]^T \in \mathbb{R}^{n+1}$. This bilinear form is sometimes referred to as the Lorentzian inner product although it is not positive-definite. We denote the norm, called *Lorentzian norm*, induced by the Lorentzian inner product by $\|\boldsymbol{x}\|_L = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_L}$. Note that $\|\boldsymbol{x}\|_L$ is either positive, zero, or positive imaginary.

We consider the following submanifold of $\mathbb{R}^{1,n}$

$$\mathbb{L}^n := \{\boldsymbol{x} = [x_0, \ldots, x_n]^T \in \mathbb{R}^{n+1} : \|\boldsymbol{x}\|_L^2 = -1, x_0 > 0\} \tag{2}$$

This is called the n-dimensional *hyperboloid model* of one sheet of a hyperbolic space defined in $\mathbb{R}^{n+1}$.

2.2. Lorentz Transformations

In the Lorentzian space, the linear isometries are called the *Lorentz transformation*, i.e. the map $\phi : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ is a Lorentz transformation if $\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle_L = \langle \boldsymbol{x}, \boldsymbol{y} \rangle_L$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}$. It is easy to see that all Lorentz transformations form a group under composition, and this group is denoted by $\mathbf{O}(1,n)$, called the *Lorentz group*. The matrix representation of $\mathbf{O}(1,n)$ in $\mathbb{R}^{n+1}$ is defined as follows. Let $J_n = \mathrm{diag}(-1, I_n)$ where I_n is the $n \times n$ identity matrix and $\mathrm{diag}(\cdot)$ denotes a diagonal matrix. Then, $\mathbf{O}(1,n)$ is defined as $\mathbf{O}(1,n) := \{A \in GL(n+1,\mathbb{R}) : AJ_nA^T = A^TJ_nA = J_n\}$, where $GL(n+1,\mathbb{R})$ is the *general linear group* of $(n+1)\times(n+1)$-invertible matrices over $\mathbb{R}$.There are a few important subgroups of $\mathbf{O}(1,n)$: (i) the subgroup $\mathbf{O}^+(1,n) := \{A \in \mathbf{O}(1,n) : a_{11} > 0\}$ is called the *positive Lorentz group*; (ii) the subgroup $\mathbf{SO}(1,n) := \{A \in \mathbf{O}(1,n) : \det(A) = 1\}$ is called the *special Lorentz group*; (iii) the subgroup $\mathbf{SO}^+(1,n) := \{A \in \mathbf{SO}(1,n) : a_{11} > 0\}$ is called the *positive special Lorentz group*. Briefly speaking, the special Lorentz group preserves the orientation, and the positive Lorentz group preserves the sign of the first entry of $\boldsymbol{x} \in \mathbb{L}^n$.

2.3. Riemannian Geometry of Hyperbolic Space

A commonly used Riemannian metric for $\mathbb{L}^n \subset \mathbb{R}^{n+1}$ is the restriction of the Lorentz inner product to the tangent space of $\mathbb{L}^n$. *Note that even though the Lorentz inner product is not positive-definite, when restricted to the tangent space of $\mathbb{L}^n$, it is positive-definite. Hence, $\mathbb{L}^n$ is a Riemannian manifold with constant negative sectional curvature.* Furthermore, the group of isometries of $\mathbb{L}^n$ is precisely $\mathbf{O}^+(1,n)$ and the group of orientation-preserving isometries is $\mathbf{SO}^+(1,n)$. We now state a few useful facts about the group of isometries that are used in this paper and refer the interested reader to [12] for details.

Fact 1. *The positive special Lorentz group $\mathbf{SO}^+(1,n)$ acts transitively on $\mathbb{L}^n$ where the group action is defined as $x \mapsto Ax$ for $x \in \mathbb{L}^n$ and $A \in \mathbf{SO}^+(1,n)$.*

Fact 2. *Let $\boldsymbol{x} = [1, 0, \ldots, 0]^T \in \mathbb{L}^n$. The isotropy subgroup $G_{\boldsymbol{x}}$ is given by*

$$\begin{aligned} G_{\boldsymbol{x}} &:= \{A \in \mathbf{SO}^+(1,n) : A\boldsymbol{x} = \boldsymbol{x}\} \\ &= \left\{ \begin{bmatrix} 1 & 0 \\ 0 & R \end{bmatrix} : R \in \mathbf{SO}(n) \right\} \cong \mathbf{SO}(n) \end{aligned} \tag{3}$$

where $\mathbf{SO}(n)$ is the group of $n \times n$ orthogonal matrices with determinant 1.

Hence, the hyperbolic space is a homogeneous Riemannian manifold and can be written as a quotient space, $\mathbb{L}^n = \mathbf{SO}^+(1,n)/\mathbf{SO}(n)$.

Fact 3 ([31]). *A Lorentz transformation $A \in \mathbf{SO}^+(1,n)$ can be decomposed using a polar decomposition and expressed as*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} c & \boldsymbol{v}^T \\ \boldsymbol{v} & \sqrt{I_n + \boldsymbol{v}\boldsymbol{v}^T} \end{bmatrix} \tag{4}$$

where $R \in \mathbf{SO}(n)$, $\boldsymbol{v} \in \mathbb{R}^n$ and $c = \sqrt{\|\boldsymbol{v}\|^2 + 1}$.

The first component in the decomposition is called a *Lorentz rotation* and the second component a *Lorentz boost*. See Figure 2 for example illustrations of the Lorentz rotation and the Lorentz boost respectively.

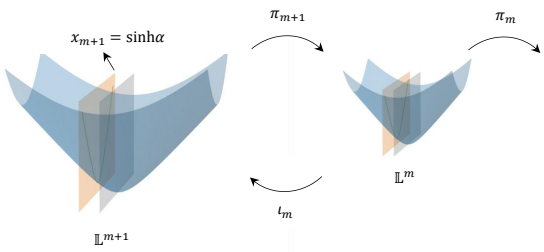


Figure 3. Illustration of NH model using the embedding ι_m in Eq. (9) of $\mathbb{L}^m$ into $\mathbb{L}^{m+1}$. The m-dimensional nested hyperboloid in $\mathbb{L}^{m+1}$ is indeed the intersection of $\mathbb{L}^{m+1}$ and an m-dimensional hyperplane.

Fact 4. *Every Lorentz transformation matrix* $A \in \mathbf{SO}^+(1,n)$ *can be decomposed into*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & P \end{bmatrix} \begin{bmatrix} \cosh\alpha & \sinh\alpha & 0 \\ \sinh\alpha & \cosh\alpha & 0 \\ 0 & 0 & I_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q^T \end{bmatrix} \tag{5}$$

where $P, Q \in \mathbf{SO}(n)$, $\alpha \in \mathbb{R}$, I_{n-1} *is the* $(n-1)\times(n-1)$ *identity matrix.*

The matrix in the middle is the Lorentz boost along the first coordinate axis. This decomposition will be very useful in the optimization problem stated in Section 3.3, equation (14).

We now conclude this section by presenting the explicit closed form formulae for the exponential map and the geodesic distance. For any $\boldsymbol{x} \in \mathbb{L}^n$ and $\boldsymbol{v} \in T_p\mathbb{L}^n$ (the tangent space of $\mathbb{L}^n$ at $\boldsymbol{x}$), the exponential map at $\boldsymbol{x}$ is given by

$$\mathrm{Exp}_{\boldsymbol{x}}(\boldsymbol{v}) = \cosh(\|\boldsymbol{v}\|_L)\boldsymbol{x} + \sinh(\|\boldsymbol{v}\|_L)\boldsymbol{v}/\|\boldsymbol{v}\|_L. \tag{6}$$

Since $\mathbb{L}^n$ is a negatively curved Riemannian manifold, its exponential map is invertible and the inverse of the exponential map, also called the Log map, is given by

$$\mathrm{Log}_{\boldsymbol{x}}(\boldsymbol{y}) = \frac{\theta}{\sinh(\theta)}(\boldsymbol{y} - \cosh(\theta)\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{L}^n$ and θ is the geodesic distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ given by $\theta = d_{\mathbb{L}}(\boldsymbol{x},\boldsymbol{y}) = \cosh^{-1}(-\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L)$.

3. Nested Hyperbolic Spaces and Networks

In this section, we first present the construction of nested hyperboloids (NHs); an illustration of the NHs are given in Figure 3. We also prove that the proposed NHs possess several nice properties, including the isometry property and the equivariance under the Lorentz transformations. Then we use the NH representations to design a novel graph convolutional network architecture, called Nested Hyperbolic Graph Convolutional Network (NHGCN).

3.1. The Nested Hyperboloid Representation

The key steps to the development of the NHs are the embedding of $\mathbb{L}^m$ into $\mathbb{L}^n$ for $m < n$ and the projection from $\mathbb{L}^n$ to $\mathbb{L}^m$. The principle is to define an embedding of the corresponding groups of isometries, $\mathbf{SO}^+(1,m)$ and $\mathbf{SO}^+(1,n)$.

First, we consider the embedding $\tilde{\iota}_m : \mathbf{SO}^+(1,m) \to \mathbf{SO}^+(1,m+1)$ defined by

$$\tilde{\iota}_m(\boldsymbol{O}) = \text{adapted-GS}\left(\boldsymbol{\Lambda}\begin{bmatrix} \boldsymbol{O} & \boldsymbol{a}^T \\ \boldsymbol{b} & c \end{bmatrix}\right) \tag{8}$$

where $\boldsymbol{O} \in \mathbf{SO}^+(1,m)$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{m+1}$, $c \neq \boldsymbol{a}^T\boldsymbol{O}^{-1}\boldsymbol{b}$, and $\Lambda \in \mathbf{SO}^+(1,m+1)$. The function adapted-GS$(\cdot)$ is an adaptation of the standard Gram-Schmidt process to orthonormalize vectors with respect to the Lorentz inner product defined earlier.

The Riemannian submersion (see [20] for the definition of a Riemannian submersion) $\pi : \mathbf{SO}^+(1,m) \to \mathbb{L}^m$ is given by $\pi(\boldsymbol{O}) = \boldsymbol{O_1}$ where $\boldsymbol{O} \in \mathbf{SO}^+(1,m)$ and $\boldsymbol{O_1}$ is the first column of O. Therefore, the induced embedding $\iota_m : \mathbb{L}^m \to \mathbb{L}^{m+1}$ is

$$\iota_m(\boldsymbol{x}) = \boldsymbol{\Lambda}\begin{bmatrix} \cosh(r)\boldsymbol{x} \\ \sinh(r) \end{bmatrix} = \cosh(r)\tilde{\boldsymbol{\Lambda}}\boldsymbol{x} + \sinh(r)\boldsymbol{v} \tag{9}$$

where $r \in \mathbb{R}$, $\boldsymbol{\Lambda} = [\tilde{\boldsymbol{\Lambda}} \quad \boldsymbol{v}] \in \mathbf{SO}^+(1,m+1)$, $\tilde{\boldsymbol{\Lambda}}$ is the first $m+1$ columns of $\boldsymbol{\Lambda}$, $\boldsymbol{v}$ is the last column of $\boldsymbol{\Lambda}$. This class of embeddings is quite general as it includes isometric embeddings as special cases.

Proposition 1. *The embedding* $\iota_m : \mathbb{L}^m \to \mathbb{L}^{m+1}$ *is isometric when* $r = 0$.

Proof. It follows directly from the definitions of the Lorentz transformation and the geodesic distance on $\mathbb{L}^m$. □

Furthermore, the embedding (9) is equivariant under Lorentz transformations.

Theorem 1. *The embedding* $\iota_m : \mathbb{L}^m \to \mathbb{L}^{m+1}$ *is equivariant under Lorentz transformations of* $\mathbf{SO}^+(1,m)$, *i.e.,* $\iota_m(\boldsymbol{R}\boldsymbol{x}) = \Psi_{\boldsymbol{\Lambda}}(\tilde{\iota}_m(\boldsymbol{R}))\iota_m(\boldsymbol{x})$ *where* $\Psi_g(h) = ghg^{-1}$.

Proof. For $\boldsymbol{x} \in \mathbb{L}^m$ and $\boldsymbol{R} \in \mathbf{SO}^+(1,m)$,

$$\begin{aligned} \iota_m(\boldsymbol{R}\boldsymbol{x}) &= \boldsymbol{\Lambda}\begin{bmatrix} \cosh(r)\boldsymbol{R}\boldsymbol{x} \\ \sinh(r) \end{bmatrix} \\ &= \boldsymbol{\Lambda}\begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \cosh(r)\boldsymbol{x} \\ \sinh(r) \end{bmatrix} \\ &= \boldsymbol{\Lambda}\begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & 1 \end{bmatrix}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}\begin{bmatrix} \cosh(r)\boldsymbol{x} \\ \sinh(r) \end{bmatrix} \\ &= \Psi_{\boldsymbol{\Lambda}}(\tilde{\iota}_m(\boldsymbol{R}))\iota_m(\boldsymbol{x}). \end{aligned}$$

□

The projection $\pi_{m+1} : \mathbb{L}^{m+1} \to \mathbb{L}^m$ corresponding to ι_m is given by,

$$\begin{aligned}\pi_{m+1}(\boldsymbol{x}) &= \frac{1}{\cosh r} J_m \tilde{\boldsymbol{\Lambda}}^T J_{m+1} \boldsymbol{x} \\ &= \frac{J_m \tilde{\boldsymbol{\Lambda}}^T J_{m+1} \boldsymbol{x}}{\|J_m \tilde{\boldsymbol{\Lambda}}^T J_{m+1} \boldsymbol{x}\|_L}\end{aligned} \tag{10}$$

for $\boldsymbol{x} \in \mathbb{L}^{m+1}$. Hence, the reconstructed point $\hat{\boldsymbol{x}} \in \mathbb{L}^{m+1}$ of $\boldsymbol{x} \in \mathbb{L}^{m+1}$ is

$$\hat{\boldsymbol{x}} = \cosh(r)\boldsymbol{\Lambda} \frac{J_m \tilde{\boldsymbol{\Lambda}}^T J_{m+1} \boldsymbol{x}}{\|J_m \tilde{\boldsymbol{\Lambda}}^T J_{m+1} \boldsymbol{x}\|_L} + \sinh(r)\boldsymbol{v}. \tag{11}$$

The unknowns $\boldsymbol{\Lambda} = [\tilde{\boldsymbol{\Lambda}} \quad \boldsymbol{v}]$ and r can then be obtained by minimizing the reconstruction error

$$L(\boldsymbol{\Lambda}, r) = \frac{1}{N} \sum_{i=1}^{N} \left(d_{\mathbb{L}}(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i)\right)^2. \tag{12}$$

The projection of $\boldsymbol{x} \in \mathbb{L}^n$ into $\mathbb{L}^m$ for $n > m$ can be obtained via the composition $\pi := \pi_{m+1} \circ \cdots \circ \pi_n$

$$\begin{aligned}\pi(\boldsymbol{x}) &= J_m \left(\prod_{i=m+1}^{n} \frac{1}{\cosh(r_i)} \tilde{\boldsymbol{\Lambda}}_i \right)^T J_n \boldsymbol{x} \\ &= \frac{J_m \boldsymbol{M}^T J_n \boldsymbol{x}}{\|J_m \boldsymbol{M}^T J_n \boldsymbol{x}\|_L}\end{aligned} \tag{13}$$

where $\boldsymbol{M} = \prod_{i=m+1}^{n} \tilde{\boldsymbol{\Lambda}}_i \in \mathbb{R}^{(n+1)\times(m+1)}$.

3.2. Nested Hyperbolic Graph Convolutional Network (NHGCN)

The Hyperbolic Graph Convolutional Network (HGCN) proposed in [6] is a generalization of Euclidean Graph Network to a hyperbolic space. There are three different layers in HGCN: feature transformation, neighborhood aggregation and non-linear activation. We use our NH representation to define a hyperbolic feature transformation, the weighted centroid w.r.t the *squared Lorentzian distance* to define the neighborhood aggregation and use a tangent ReLU activation. This leads to a novel HGCN architecture. Figure 4 depicts the HGCN architecture. Each of the three distinct layers are described in detail below.

Hyperbolic Feature Transformation: Given $\boldsymbol{x} \in \mathbb{L}^n$, the hyperbolic feature transformation is defined using (13) as follows

$$\boldsymbol{y} = \frac{\boldsymbol{W}\boldsymbol{x}}{\|\boldsymbol{W}\boldsymbol{x}\|_L} \qquad \text{s.t. } \boldsymbol{W} J_n \boldsymbol{W}^T = J_m \tag{14}$$

where $\boldsymbol{W} \in \mathbb{R}^{(m+1)\times(n+1)}$. It is easy to prove that $\boldsymbol{y} \in \mathbb{L}^m$.

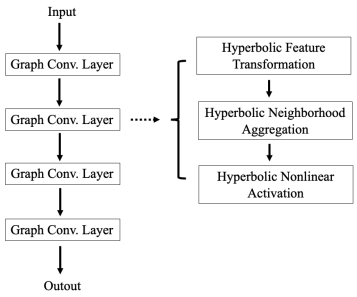


Figure 4. The HGCN Architecture

At the l-th layer, the inputs are the hyperbolic representation $\boldsymbol{x}_i^{l-1}$ from the previous layer and the feature transformation matrix is $\boldsymbol{W}^l$. The intermediate hyperbolic representation of i-th node is computed as follows

$$\boldsymbol{x}_i^l = \frac{\boldsymbol{W}^l \boldsymbol{x}_i^{l-1}}{\|\boldsymbol{W}^l \boldsymbol{x}_i^{l-1}\|_L} \qquad \text{s.t. } \boldsymbol{W}^l J_{n_{l-1}} {\boldsymbol{W}^l}^T = J_{n_l} \tag{15}$$

Hyperbolic Neighborhood Aggregation: In GCNs, the neighborhood aggregation is used to combine neighboring features by computing the weighted centroid of these features. The weighted centroid in hyperbolic space of a point set $\{\boldsymbol{x}_i\}_{i=1} \in \mathbb{L}^n$ is obtained using the weighted Fréchet mean. However, it does not have closed form expression in hyperbolic space. We use hyperbolic neighborhood aggregation proposed in [7, 49], where aggregated representation for a node $\boldsymbol{x}_i^l$ at l-th layer is the weighted centroid $\boldsymbol{\mu}_i^l$ of its neighboring nodes $\{\boldsymbol{x}_j^l\}_{j=1}^p \in \mathbb{L}^{n_l}$ w.r.t *squared Lorentzian distance*, namely

$$\boldsymbol{\mu}_i^l = \arg\min_{\boldsymbol{\mu}^l \in \mathbb{L}^{n_l}} \sum_{j=1}^{p} \nu_j^l d_{\mathbb{L}}^2(\boldsymbol{x}_j^l, \boldsymbol{\mu}_i^l) \tag{16}$$

where ν_j^l is the weight for $\boldsymbol{x}_j^l$ and $d_{\mathbb{L}}^2(\boldsymbol{x}, \boldsymbol{y}) = -1 - \langle \boldsymbol{x}, \boldsymbol{y} \rangle_L$ is the squared Lorentzian distance [37]. Authors in [27] proved that this problem has closed form solution given by,

$$\boldsymbol{\mu}_i^l = \frac{\sum_{j=1}^{p} \nu_j^l \boldsymbol{x}_j^l}{|\|\sum_{j=1}^{p} \nu_j^l \boldsymbol{x}_j^l\|_L|}. \tag{17}$$

Hyperbolic Nonlinear Activation: A nonlinear activation is required in our network since the feature transform is a linear operation. We choose to apply tangent ReLU to prevents our multi-layer network from collapsing into a single layer network. The tangent ReLU in the hyperbolic space is defined as,

$$\sigma(\boldsymbol{x}_i^l) = \mathrm{Exp}_{\mathbf{0}}(\mathrm{ReLU}(\mathrm{Log}_{\mathbf{0}}(\boldsymbol{x}_i^l))). \tag{18}$$

Here $\mathbf{0} = [1, 0, \ldots, 0]^T \in \mathbb{L}^{n_l}$ (correspond to the origin in the Poincaré model) is chosen as the base point to define the anchor point in the tangent ReLU.

3.3. Optimization

In this section, we will explain how to update parameters in network, i.e. transformation matrix $\boldsymbol{W}$ in (14). Instead of updating $\boldsymbol{W}$ directly, we find an alternative way by decomposing $\boldsymbol{W}$ into three matrices using (5). More specifically, we write

$$\boldsymbol{W} = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{P} \end{bmatrix} \begin{bmatrix} \cosh\alpha & \sinh\alpha & 0 \\ \sinh\alpha & \cosh\alpha & 0 \\ 0 & 0 & I_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q^T \end{bmatrix}$$

where $Q \in \mathbf{SO}(n), \alpha \in \mathbb{R}$ and $\widetilde{P}$ is the first m rows of a $P \in \mathbf{SO}(n)$ which is from a *Stiefel manifold* [9]. We can now update these factors sequentially in the optimization (see supplement material for details).

4. Experiments

In this section, we will first evaluate the NH model as a dimensionality reduction method compared with HoroPCA, tPCA and EPGA. We show that the proposed NH model outperforms all of these method on both synthetic data and real data in terms of reconstruction error. Then, we apply the proposed NHGCN to the problems of link prediction and node classification on four graph data sets described in [6]. Our method yields results that are better or comparable to existing hyperbolic graph networks. The implementations[1] are based on Pymanopt [26] and GeoTorch [28] for dimensionality reduction and NHGCN respectively.

4.1. Dimensionality Reduction in Hyperbolic Space

First we present synthetic data experiments followed by experiments on real data.

Synthetic Experiments As a dimensionality reduction method, we compare the NH model with three other competing methods: tPCA, EPGA, and HoroPCA. Note that the first two are applicable on any Riemannian manifolds and HoroPCA is proposed specifically for hyperbolic spaces as is our NH model. The major difference between NH and the aforementioned methods is that NH does not require the fitted submanifold to pass through the FM whereas the others do. This extra requirement can sometimes lead to failure in capturing the data trend as shown in Figure 5.

Apart from visual inspection, we use the reconstruction error as a measure of the goodness of fit. To see how NH performs in comparison to others under different levels of noise, we generate synthetic data from the *wrapped normal distribution* [30] on $\mathbb{L}^{10}$ with variance ranging from 0.2 to 2. Then we apply different dimensionality reduction methods to reduce the dimension down to 2. The result is shown in Figure 6. The results for EPGA and NH are essentially the same since, for wrapped normal distribution, the data is distributed symmetrically around the FM and hence the assumption of submanifold passing through the FM is valid here. Even in this case, we observe a significant improvement of NH over tPCA and HoroPCA especially in the large variance scenario. The main reasons are that (i) tPCA uses local linearization which would lead to inaccuracies when the data is not tightly clustered around the FM and (ii) the HoroPCA seeks to maximize the projected variance on the submanifold, which, as is well known, not equivalent to minimizing the reconstruction error. There is a clear justification for the choice of using reconstruction error as the objective function since, *we want a good approximation of the original data with the lower-dimensional representation*. For additional experimental results, please see supplementary material.

[1] https://github.com/cvgmi/Nested-Hyperbolic-DimReduc-and-HNN.

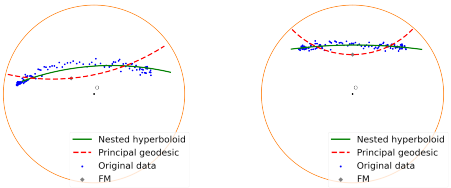


Figure 5. Synthetic data in hyperbolic space along with principal geodesic obtained using tPCA and NH. NH captures the data trend better as it is not restricted to pass through the FM.

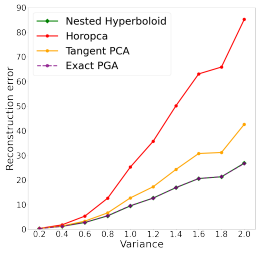


Figure 6. Reconstruction errors for $\mathbb{L}^{10}$ to $\mathbb{L}^{2}$. The data is generated from *wrapped normal distributions* [30] with variances ranging from 0.2 to 2.

Hyperbolic Embeddings of Trees For real data experiments, we consider reducing the dimensionality of trees that are embedded into a hyperbolic space. We validate our method on the four datasets described in [38] including (i) a fully balanced tree, (ii) a phylogenetic tree, (iii) a biological graph comprising of diseases' relationships, and (iv) a graph of Computer Science (CS) Ph.D. advisor-advisee relationships. We also create two additional datasets by removing some edges in the balanced tree dataset. We apply the method in [14] to embed the tree datasets into a Poincaré ball of dimension 10 and then apply our NH model

Datasets	balancedtree	unbalanced1	unbalanced2	phylo tree	diseasome	ca-CSphd
tPCA	5.75 (1.5)	4.98 (0.001)	4.86 (0.001)	121.19 (0.001)	21.53 (0.004)	71.67 (0.4)
HoroPCA	7.80 ±0.06 (2.5)	6.51±0.28 (2.4)	7.35±0.61 (2.3)	108.62±9.20 (78)	26.94±0.99 (136)	87.99±4.69 (500)
EPGA	4.01±0.76 (2.2)	3.23±0.08 (1.8)	3.33±0.46 (1.9)	25.93±0.99 (2.6)	9.72±0.36 (3.0)	22.98±0.23 (5.9)
NH(Ours)	**3.35**±0.05 (4.1)	**3.10**±0.01 (3.5)	**3.22**±0.06 (2.5)	**24.11**±0.68 (25)	**9.18**±0.10 (31)	**22.68**±0.40 (87)

Table 1. Reconstruction errors from $\mathbb{L}^{10}$ to $\mathbb{L}^2$. The numbers depicted are: mean error ± standard dev. of error. Numbers in bold indicate the method with the smallest errors while underlined numbers indicate the second best results. Numbers in parentheses indicate the running time in seconds on a Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30GHz.

	Disease		Airport		PubMed		Cora	
Task	LP	NC	LP	NC	LP	NC	LP	NC
GCN [25]	64.7±0.5	69.7±0.4	89.3±0.4	81.4±0.6	91.1±0.5	78.1±0.2	90.4±0.2	81.3±0.3
GAT [45]	69.8±0.3	70.4±0.4	90.5±0.3	81.5±0.3	91.2±0.1	79.0±0.3	93.7±0.1	83.0±0.7
SAGE [16]	65.9±0.3	69.1±0.6	90.4±0.5	82.1±0.5	86.2±1.0	77.4±2.2	85.5±0.6	77.9±2.4
SGC [46]	65.1±0.2	69.5±0.2	89.8±0.3	80.6±0.1	94.1±0.0	78.9±0.0	91.5±0.1	81.0±0.1
HGCN [6]	90.8±0.3	74.5±0.9	96.4±0.1	90.6±0.2	96.3±0.0	80.3±0.3	92.9±0.1	79.9±0.2
H2H-GCN [8]	**97.0**±0.3	88.6±1.7	96.4±0.1	89.3±0.5	**96.9**±0.0	79.9±0.5	**95.0**±0.0	82.8±0.4
HYBONET [7]	96.3±0.3	**94.5**±0.8	97.0±0.2	**92.5**±0.9	96.4±0.1	77.9±1.0	94.3±0.3	81.3±0.9
LGCN [49]	96.6±0.6	84.4±0.8	-	-	96.6±0.1	78.6±0.7	93.6±0.4	**83.3**±0.7
NHGCN(Ours)	92.8±0.2	91.7±0.7	**97.2**±0.3	92.4±0.7	**96.9**±0.1	**80.5**±0.0	93.6±0.2	80.3±0.8

Table 2. Area under the ROC test results (%) for LP, and F1 scores (%) for NC. Results of other networks are obtained from the original papers, and in [49], the authors did not test their network on the Airport dataset.

along with other competing methods to reduce the dimension down to 2. The results are reported in Table 1. In Table 1, we report the means, the standard deviations of the reconstruction errors and the running time in seconds for EPGA, HoroPCA and NH respectively. From the table, it is evident that our method performs the best and is better than HoroPCA, the SOTA. Specifically, HoroPCA is worse than the tPCA and EPGA in terms of reconstruction error, though it yields higher explained variance as shown in [5]. The reason might be that HoroPCA seeks projections that maximize the explained variance which is not equivalent to minimizing the reconstruction error in the Riemannian manifold case.

4.2. Nested Hyperbolic Graph Networks

To evaluate the power of the proposed NHGCN, we apply it to problems of link prediction (LP) and node classification (NC). We use four public domain datasets: Disease [6], Airport [6], PubMed [32], and Cora [40]. We compare our NHGCN with many other graph neural networks and the results are reported in Table 2. For the LP, we report the means and the standard deviation of the area under the receiver operating characteristic (ROC) curve on the test data; for the problem of NC, we report the mean and the standard deviation of the F1 scores. As evident from the table, our results are comparable to the state-of-the-art (SOTA) and in three cases better. A noteworthy point about our model is that all the operations used in the model are intrinsic to $\mathbb{L}^n$ unlike the others in Table 2. Intrinsic operations by definition should yield better accuracy. Thus, we attribute the lower accuracy of our model in Table 2 to the sub-optimal optimization approach used here. This optimization problem is on a semi-Riemannian manifold, an open problem that will be addressed in future work.

5. Conclusion

In this paper, we presented a novel dimensionality reduction technique in hyperbolic spaces called the NH representation. NH representation was constructed using a projection operator that was shown to yield isometric embeddings and further was shown to be equivariant to the isometry group admitted by the hyperbolic space. Further, we empirically showed that it yields lower reconstruction error compared to the state-of-the-art (HorroPCA, EPGA, tPCA). Using the NH representation, we developed a novel fully HGCN and tested it on several data sets. Our NHGCN was shown to achieve comparable to superior performance w.r.t. several competing methods.

Acknowledgements. This research was in part funded by the NSF grant IIS-1724174, the NIH NINDS and NIA via R01NS121099 to Vemuri and the MOST grant 110-2118-M-002-005-MY3 to Yang.

References

[1] Monami Banerjee, Rudrasis Chakraborty, and Baba C Vemuri. Sparse exact pga on riemannian manifolds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5010–5018, 2017. 2

[2] Herbert Busemann. The geometry of geodesics. *Pure and Applied Mathematics*, 1955. 2

[3] James W. Cannon, William J. Floyd, Richard Kenyon, and Walter R. Parry. *Hyperbolic Geometry*, volume 31. MSRI Publications, 1997. 2, 3, 4

[4] Rudrasis Chakraborty, Dohyung Seo, and Baba C Vemuri. An efficient exact-pga algorithm for constant curvature manifolds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3976–3984, 2016. 2

[5] Ines Chami, Albert Gu, Dat P Nguyen, and Christopher Ré. Horopca: Hyperbolic dimensionality reduction via horospherical projections. In *International Conference on Machine Learning*, pages 1419–1429. PMLR, 2021. 2, 8

[6] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in Neural Information Processing Systems*, 32:4868–4879, 2019. 3, 6, 7, 8

[7] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. *arXiv preprint arXiv:2105.14686*, 2021. 2, 3, 6, 8

[8] Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 154–163, 2021. 3, 8

[9] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998. 7

[10] P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004. 1

[11] Maurice Fréchet. Les 'el´ements al´eatoires de nature quelconque dans un espace distanci´e. *Ann. Inst. H. Poincar´e*, 10:215–310, 1948. 1

[12] Jean Gallier and Jocelyn Quaintance. Notes on differential geometry and lie groups. *University of Pennsylvannia*, 4:3–1, 2012. 4

[13] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in Neural Information Processing Systems 31*, pages 5345–5355, 2019. 3

[14] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*, 2018. 7

[15] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2018. 3

[16] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017. 8

[17] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):48–62, 2018. 2

[18] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989. 2

[19] Søren Hauberg. Principal curves on riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1915–1921, 2015. 2

[20] Sigurdur Helgason. *Differential geometry, Lie groups, and symmetric spaces*. Academic Press, 1979. 5

[21] Stephan Huckemann, Thomas Hotz, and Axel Munk. Intrinsic shape analysis: Geodesic pca for riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, pages 1–58, 2010. 2

[22] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016. 1

[23] Sungkyu Jung, Ian L Dryden, and James Stephen Marron. Analysis of principal nested spheres. *Biometrika*, 99(3):551–568, 2012. 2, 3

[24] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020. 3

[25] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017. 8

[26] Niklas Koep and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17:1–5, 2016. 7

[27] Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. Lorentzian distance learning for hyperbolic representations. In *International Conference on Machine Learning*, pages 3672–3681. PMLR, 2019. 6

[28] Mario Lezcano-Casado. Trivializations for gradient-based optimization on manifolds. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 9154–9164, 2019. 7

[29] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems*, 32:8230–8241, 2019. 3

[30] Emile Mathieu. Charline le lan, chris j maddison, ryota tomioka, and yee whye teh. continuous hierarchical representations with poincaré variational auto-encoders. *Advances in Neural Information Processing Systems*, pages 12544–12555, 2019. 3, 7

[31] Valter Moretti. The interplay of the polar decomposition theorem and the lorentz group. *arXiv preprint math-ph/0211047*, 2002. 4

[32] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and UMD EDU. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, volume 8, page 1, 2012. 8

[33] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30:6338–6347, 2017. 3

[34] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3779–3788. PMLR, 2018. 2

[35] Jiwoong Park, Junho Cho, Hyung Jin Chang, and Jin Young Choi. Unsupervised hyperbolic representation learning via message passing auto-encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5516–5526, 2021. 3

[36] Xavier Pennec. Barycentric subspace analysis on manifolds. *The Annals of Statistics*, 46(6A):2711–2746, 2018. 2

[37] John G Ratcliffe. *Foundations of Hyperbolic Manifolds*, volume 149. Springer, 2 edition, 2006. 3, 6

[38] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018. 3, 7

[39] Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011. 3

[40] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008. 8

[41] Ryohei Shimizu, YUSUKE Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. In *International Conference on Learning Representations*, 2020. 3

[42] Stefan Sommer, François Lauze, Søren Hauberg, and Mads Nielsen. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In *European conference on computer vision*, pages 43–56. Springer, 2010. 2

[43] Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*, 2018. 3

[44] Abraham A Ungar. Gyrovector spaces and their differential geometry. *Nonlinear Funct. Anal. Appl*, 10(5):791–834, 2005. 3

[45] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. 8

[46] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019. 8

[47] Chun-Hao Yang and Baba C Vemuri. Nested grassmanns for dimensionality reduction with applications to shape analysis. In *International Conference on Information Processing in Medical Imaging*, pages 136–149. Springer, 2021. 2, 3

[48] Miaomiao Zhang and Tom Fletcher. Probabilistic principal geodesic analysis. *Advances in Neural Information Processing Systems*, 26:1178–1186, 2013. 2

[49] Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In *Proceedings of the Web Conference 2021*, pages 1249–1261, 2021. 6, 8