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
metric positive definite (SPD) matrices, authors in [17] pro-
be geodesic. Similarly, for the manifold $P_{NS}$ is more general than $PGA$ in that $P_{NS}$ does not have to
the embedding however is not necessarily isometric. Hence, $(P_{NS})$ by embedding an
in [23] where authors proposed the
spaces form a nested hierarchy. This idea was exploited
appropriate for certain applications.

This is evident from
significant difference in performance.

inaccuracies. Subsequently, [42] presented the Exact PGA
algorithm, which does not use any linear approx-
imation. However, EPGA is computationally expensive as it requires two non-linear optimizations steps per iteration
(projection to the geodesic submanifold and finding the new
goesic direction such that the reconstruction error is min-
imized). Later, authors in [4] developed a version of EPGA
for constant sectional curvature manifolds, namely the hy-
persphere 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 parameteriza-
tion of unexplained variance. Another useful dimensional-
ity 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 lin-
ear subspaces, i.e., the reduced dimensional principal sub-
spaces 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] pro-
posed a geometry-aware dimensionality reduction by pro-
jecting 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 man-
ifolds, 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 hy-
perbolic 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 facil-
itate 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 bet-
ter 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 gener-
alize the notion of direction and the coordinates in a given
direction using ideal points (points at infinity) and the Buse-
mann 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 move points along a horo-
sphere. 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 be-
tween 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 di-
ensional representations obtained by our method in compari-
son 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 dimen-
sionality 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].
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 Lorentz space \(\mathbb{R}^{1,n}\) is the Euclidean space \(\mathbb{R}^{n+1}\) equipped with a bilinear form

\[
\langle x, y \rangle_L = -x_0y_0 + x_1y_1 + \cdots + x_ny_n
\]

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

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

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

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 \(\phi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}\) is a Lorentz transformation if \(\langle \phi(x), \phi(y) \rangle_L = \langle x, y \rangle_L\) for any \(x, y \in \mathbb{R}^{n+1}\). It is easy to see that all Lorentz transformations form a group under composition, and this group is denoted by \(O(1, n)\), called the Lorentz group. The matrix representation of \(O(1, n)\) in \(\mathbb{R}^{n+1}\) is defined as follows. Let \(J_n = \text{diag}(-1, I_n)\) where \(I_n\) is the \(n \times n\) identity matrix and \(\text{diag}(\cdot)\) denotes a diagonal matrix. Then, \(O(1, n)\) is defined as \(O(1, n) := \{ A \in GL(n+1, \mathbb{R}) : AJ_nA^T = A^T J_nA = J_n \} \), 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 \(O(1, n)\): (i) the subgroup \(O^+(1, n) := \{ A \in O(1, n) : a_{11} > 0 \}\) is called the positive Lorentz group; (ii) the subgroup \(SO(1, n) := \{ A \in \text{O}(1, n) : \text{det}(A) = 1 \}\) is called the special Lorentz group; (iii) the subgroup \(SO^+(1, n) := \{ A \in \text{SO}(1, n) : a_{11} > 0 \}\) 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 \(x \in \mathbb{L}^n\).

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 \(O^+(1, n)\) and the group of orientation-preserving isometries is \(SO^+(1, n)\). 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 \(SO^+(1, n)\) acts transitively on \(\mathbb{L}^n\) where the group action is defined as \(x \mapsto Ax\) for \(x \in \mathbb{L}^n\) and \(A \in SO^+(1, n)\).

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

\[
G_x := \{ A \in SO^+(1, n) : Ax = x \}
\]

\[
= \left\{ \begin{bmatrix} 1 & 0 \\ 0 & R \end{bmatrix} : R \in SO(n) \right\} \approx SO(n)
\]

where \(SO(n)\) 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, \(\mathbb{L}^n = SO^+(1, n)/SO(n)\).

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

\[
A = \begin{bmatrix} 1 & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} c & v^T \\ \sqrt{v^T v + 1} & u \end{bmatrix}
\]

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

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.
Every Lorentz transformation matrix \( A \in SO^+(1, n) \) can be decomposed into

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

where \( P, Q \in SO(n), \alpha \in \mathbb{R}, I_{n-1} \) 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 \( x \in \mathbb{L}^n \) and \( v \in T_p \mathbb{L}^n \) (the tangent space of \( \mathbb{L}^n \) at \( x \)), the exponential map at \( x \) is given by

\[
\exp_x(v) = \cosh(\|v\|_L)x + \sinh(\|v\|_L)v/\|v\|_L.
\]

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

\[
\log_x(y) = \frac{\theta}{\sinh(\theta)}(y - \cosh(\theta)x)
\]

where \( x, y \in \mathbb{L}^n \) and \( \theta \) is the geodesic distance between \( x \) and \( y \) given by

\[
\theta = d_L(x, y) = \cosh^{-1}(-\langle x, y \rangle_L).
\]

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 \( \mathbb{L}^{m+1} \) 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, \( SO^+(1, m) \) and \( SO^+(1, n) \).

First, we consider the embedding \( \iota_m : SO^+(1, m) \to SO^+(1, m+1) \) defined by

\[
\iota_m(O) = \text{adapted-GS}(\Lambda)
\]

where \( O \in SO^+(1, m), a, b \in \mathbb{R}^{m+1}, c \neq a^TO^{-1}b, \) and \( \Lambda \in SO^+(1, m+1) \). The function adapted-GS(·) 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) \( \pi : SO^+(1, m) \to \mathbb{L}^m \) is given by \( \pi(O) = O_1 \) where \( O \in SO^+(1, m) \) and \( O_1 \) is the first column of \( O \). Therefore, the induced embedding \( \iota_m : \mathbb{L}^m \to \mathbb{L}^{m+1} \) is

\[
\iota_m(x) = \Lambda \begin{bmatrix}
\cosh(r)x \\
\sinh(r)
\end{bmatrix} = \cosh(r)\tilde{\Lambda}x + \sinh(r)v
\]

where \( r \in \mathbb{R}, \Lambda = [\tilde{\Lambda} v] \in SO^+(1, m+1), \tilde{\Lambda} \) is the first \( m+1 \) columns of \( \Lambda \), \( v \) is the last column of \( \Lambda \). 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 \( SO^+(1, m) \), i.e.,

\[
\iota_m(Rx) = \Psi_{\Lambda}(\iota_m(R))\iota_m(x)
\]

where \( \Psi_{\Lambda}(h) = ghg^{-1} \).

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

\[
\iota_m(Rx) = \Lambda \begin{bmatrix}
\cosh(r)Rx \\
\sinh(r)
\end{bmatrix}
\]

\[
= \Lambda \begin{bmatrix}
R & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
\cosh(r)x \\
\sinh(r)
\end{bmatrix}
\]

\[
= \Lambda \begin{bmatrix}
R & 0 \\
0 & 1
\end{bmatrix} \Lambda^{-1} \begin{bmatrix}
\cosh(r)x \\
\sinh(r)
\end{bmatrix}
\]

\[
= \Psi_{\Lambda}(\iota_m(R))\iota_m(x).
\]

360
The projection \( \pi_{m+1} : \mathbb{L}^{m+1} \rightarrow \mathbb{L}^m \) corresponding to \( \ell_m \) is given by,

\[
\pi_{m+1}(x) = \frac{1}{\cosh r} J_m \Lambda^T J_{m+1} x
= \frac{J_m \Lambda^T J_{m+1} x}{\| J_m \Lambda^T J_{m+1} x \|_L}
\]

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

\[
\hat{x} = \cosh(r) \Lambda - \frac{J_m \Lambda^T J_{m+1} x}{\| J_m \Lambda^T J_{m+1} x \|_L} + \sinh(r) v. \quad (11)
\]

The unknowns \( \Lambda = [\Lambda \ v] \) and \( r \) can then be obtained by minimizing the reconstruction error

\[
L(\Lambda, r) = \frac{1}{N} \sum_{i=1}^{N} (d_L(x_i, \hat{x}_i))^2. \quad (12)
\]

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

\[
\pi(x) = J_m \left( \prod_{i=m+1}^{n} \frac{1}{\cosh(r_i) \Lambda_i} \right)^T J_n x
= \frac{J_m M^T J_n x}{\| J_m M^T J_n x \|_L}
\]

where \( M = \prod_{i=m+1}^{n} \Lambda_i \in \mathbb{R}^{(n+1) \times (m+1)}. \)

### 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 \( x \in \mathbb{L}^n \), the hyperbolic feature transformation is defined using (13) as follows

\[
y = \frac{W x}{\| W x \|_L} \quad \text{s.t. } W J_n W^T = J_m \quad (14)
\]

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

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

\[
x_i^l = \frac{W^l x_i^{l-1}}{\| W^l x_i^{l-1} \|_L} \quad \text{s.t. } W^l J_{n_i-1} W^l^T = J_{n_i} \quad (15)
\]

**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 \( \{ x_i \}_{i=1}^{n} \in \mathbb{L}^n \) 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 \( x_i^l \) at \( l \)-th layer is the weighted centroid \( \mu_i^l \) of its neighboring nodes \( \{ x_j^l \}_{j=1}^p \in \mathbb{L}^n \) w.r.t squared Lorentzian distance, namely

\[
\mu_i^l = \arg \min_{\mu^l \in \mathbb{L}^n} \sum_{j=1}^{p} \nu_j^l d_L^2(x_i^l, \mu_i^l) \quad (16)
\]

where \( \nu_j^l \) is the weight for \( x_j^l \) and \( d_L^2(x, y) = -1 - \langle x, y \rangle_L \) is the squared Lorentzian distance [37]. Authors in [27] proved that this problem has closed form solution given by,

\[
\mu_i^l = \frac{\sum_{j=1}^{p} \nu_j^l x_j^l}{\| \sum_{j=1}^{p} \nu_j^l x_j^l \|_L}. \quad (17)
\]

**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,

\[
\sigma(x_i^l) = \text{Exp}_0(\text{ReLU}(\text{Log}_0(x_i^l))) \quad (18)
\]

Here \( 0 = [1, 0, \ldots, 0]^T \in \mathbb{L}^{n_i} \) (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 $W$ in (14). Instead of updating $W$ directly, we find an alternative way by decomposing $W$ into three matrices using (5). More specifically, we write

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

where $Q \in SO(n)$, $\alpha \in \mathbb{R}$ and $\hat{P}$ is the first $m$ rows of a $P \in SO(n)$ 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 methods 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 results yield that are better or comparable to existing hyperbolic graph networks. The implementations 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. We show that the proposed NH model outperforms all of these methods 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 results yield that are better or comparable to existing hyperbolic graph networks. The implementations 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.

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 $L^{10}$ to $L^2$. The data is generated from wrapped normal distributions [30] with variances ranging from 0.2 to 2.

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.

**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.

---

1https://github.com/cvmm/Nested-Hyperbolic-DimReduct-and-HNN.
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 IPCA 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 $L^n$ 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 (HoroPCA, EPGA, tPCA). Using the NH representation, we developed a novel fully HGSCN 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


