Expanding the Hyperbolic Kernels: A Curvature-aware Isometric Embedding View

Meimei Yang[1,2] , **Pangfei Fang**[1,2] , **Hui Xue**[1,2*]

[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]MOE Key Laboratory of Computer Network and Information Integration (Southeast University), China
{meimeiyang,fangpengfei,hxue}@seu.edu.cn

Abstract

Modeling data relation as a hierarchical structure has proven beneficial for many learning scenarios, and the hyperbolic space, with negative curvature, can encode such data hierarchy without distortion. Several recent studies also show that the representation power of the hyperbolic space can be further improved by endowing the kernel methods. Unfortunately, the known kernel methods, developed in hyperbolic space, are limited by the adaptation capacity or distortion issues. This paper addresses the issues through a novel embedding function. To this end, we propose a curvature-aware isometric embedding, which establishes an isometry from the Poincaré model to a special reproducing kernel Hilbert space (RKHS). Then we can further define a series of kernels on this RKHS, including several positive definite kernels and an indefinite kernel. Thorough experiments are conducted to demonstrate the superiority of our proposals over existing-known hyperbolic and Euclidean kernels in various learning tasks, e.g., graph learning and zero-shot learning.

1 Introduction

This paper studies a curvature-aware isometric embedding to map the hyperbolic representations into an intermediate reproducing kernel Hilbert space (RKHS), where a set of kernels can be further developed.

In recent years, the hyperbolic space has shown its superiority to encode the data hierarchy, since the volume of hyperbolic space increases exponentially with radius [Ganea *et al.*, 2018a; Ganea *et al.*, 2018b]. This property enables the hyperbolic space to encode the graph-structured or treelikeness data with arbitrarily low distortion in case of low dimension, and benefits a lot of data embeddings, e.g., language data [Ganea *et al.*, 2018a], visual data [Khrulkov *et al.*, 2020; Fang *et al.*, 2023], graph data [Cho *et al.*, 2019], etc.

To enjoy the representation power of the hyperbolic space, various learning paradigms, including hyperbolic neural networks [Ganea *et al.*, 2018b; Gulcehre *et al.*, 2018; Shimizu *et al.*, 2020] and hyperbolic kernel machines [Cho *et al.*, 2019; Fang *et al.*, 2021], are developed in the hyperbolic space, gaining substantial improvement over many AI applications. In [Ganea *et al.*, 2018b], the hyperbolic geometry is first integrated into the deep neural network as the embedding space, and some essential hyperbolic neural layers are developed to work with the hyperbolic geometry. To make the hyperbolic space benefit from the rich kernel theories, Cho *et al.* first propose a hyperbolic polynomial kernel, an indefinite kernel defined in the Lorentz model [Cho *et al.*, 2019]. The proposed kernel improves the SVM classification over graph and language data. As the complementary concept of the indefinite kernel, the positive definite (PD) kernels are further studied in [Fang *et al.*, 2021]. Fang *et al.* leverage the identity tangent space to approximate the hyperbolic space and propose a set of PD kernels in the tangent space. Those kernels show superior performance on visual recognition tasks. However, the fixed curvature of hyperbolic space limits the adaptation capacity of the kernel [Cho *et al.*, 2019], or the employment of the identity tangent space causes the distortion of hyperbolic data [Fang *et al.*, 2021].

The work in [Arcozzi *et al.*, 2007] proposes a kernel mapping, which realizes the isometry from the unit Poincaré model to a special RKHS. In other words, such a mapping is able to address the distortion issue of embedding the hyperbolic data in Euclidean space[1]. Also, many existing successful practices of hyperbolic space also show that the Poincaré model can adapt different data by tuning the curvature [Ganea *et al.*, 2018b; Chami *et al.*, 2019; Khrulkov *et al.*, 2020; Fang *et al.*, 2021]. The analysis arises a question: *can we develop some valid kernels that can jointly preserve the data structure in the hyperbolic space, while fitting the data adaptively?*

We answer this question by developing a curvature-aware isometric embedding, yielding the isometry to map data from the Poincaré model to an RKHS. Then this RKHS, with Euclidean structure, allows us to construct a series of new hyperbolic kernel functions, including PD and indefinite kernels. Enjoying the property of the proposed curvature-aware isometric embedding, our kernels can reduce the distortion of working on hyperbolic data, and also improve the adaptabil-

*Corresponding author.

[1]The RKHS can be understood as a high-dimensional (or even infinite-dimensional) Euclidean space.

ity to different data. These advantages generalize our kernels to different practical applications.

The **contributions** of this paper are summarized as follows:

- We propose a curvature-aware isometric embedding that realizes an isometry between the Poincaré model and an RKHS. This mapping can preserve the rich structure of the hyperbolic data, and adapt to different data.
- On top of the embedding function, we develop a series of hyperbolic kernels, including PD kernels and an indefinite kernel. The PD kernels have the curvature-aware hyperbolic linear kernel, the curvature-aware hyperbolic polynomial kernel, the curvature-aware hyperbolic RBF kernel, and the curvature-aware hyperbolic Laplacian kernel, while the indefinite kernel is called the curvature-aware hyperbolic Sigmoid kernel.
- Thorough experiments are performed on different learning scenarios (e.g., graph learning and zero-shot learning) to evaluate the superiority of the proposed kernels, and the adaptation capacity to different data.

2 Preliminaries

2.1 Notations

Throughout the paper, we use $\mathbb{C}^n$, $\mathbb{P}_c^n$, $\mathbb{D}_c^n$ to denote the n-dimensional complex space, n-dimensional Poincaré model with curvature $-c$, n-dimensional open ball with radius $1/\sqrt{c}$. It is worthy noted that we omit the n or c when $n = 1$ or $c = 1$ for simplicity. We also denote $\mathcal{H}$ as an RKHS and $\mathcal{K}$ as a reproducing kernel Kreĭn space (RKKS). The matrices, vectors and scalars are denoted by bold capital letters (e.g., $\boldsymbol{X}$), bold lower-case letters (e.g., $\boldsymbol{x}$) and thin letters (e.g., x), respectively.

2.2 Hyperbolic Geometry

The hyperbolic space is a Riemannian manifold with constant negative curvature [Ratcliffe, 1994]. Five isometric models, including the Poincaré model, the Lorentz model, the Klein model, the Half-space model and the Hemisphere mode, are used to model the hyperbolic space [Cannon *et al.*, 1997]. In this work, we use Poincaré model, as in many existing works [Nickel and Kiela, 2017; Ganea *et al.*, 2018b; Fang *et al.*, 2021]. The gyrovector space provides an elegant framework to work with hyperbolic space analytically [Ungar, 2022], and the Möbius gyrovector space comes in handy to work with Poincaré model [Ungar, 1998].

Poincaré model. The n-dimensional Poincaré model can be characterized as

$$\mathbb{P}_c^n = \{\boldsymbol{z} \in \mathbb{C}^n : c\|\boldsymbol{z}\|^2 < 1, c > 0\}, \tag{1}$$

where $\mathbb{C}^n$ and $-c$ represent the n-dimensional complex space and curvature of the Poincaré model, respectively.

Möbius Gyrovector Space. A Möbius gyrovector space is a gyrometric space with gyrometric, known as the Möbius gyrodistance function which obeys the gyrotriangle inequality [Ungar, 2022]. It provides a set of algebraic formalism to work for the Poincaré model. That said, one can enjoy from the Möbius gyrovector space to perform the basic vector operations, e.g., addition, subtraction etc. For any two points $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{P}_c^n$, the Möbius addition is given by

$$\boldsymbol{z}_i \oplus_c^n \boldsymbol{z}_j = \frac{\boldsymbol{z}_i + \boldsymbol{z}_j}{1 + c\langle \boldsymbol{z}_j, \boldsymbol{z}_i \rangle}. \tag{2}$$

The Möbius gyrodistance can be deduced as

$$d(\boldsymbol{z}_i, \boldsymbol{z}_j) = \|\boldsymbol{z}_j \ominus_c^n \boldsymbol{z}_i\| = \|\boldsymbol{z}_j \oplus_c^n (-\boldsymbol{z}_i)\|, \tag{3}$$

where $\ominus_c^n$ is the Möbius subtraction.

2.3 Isometry from the Unit Poincaré Model to an RKHS

Isometry is a mapping that preserves the distance (or metric) between two metric spaces [Coxeter and Greitzer, 1967]. Concretely, the equivalence between pseudo-hyperbolic distance of the unit Poincaré model and the metric, defined on an RKHS, establish the isometry. In this section, we will show the details of such an isometry realized by a kernel mapping.

Pseudo-hyperbolic Distance

The complex open unit ball $\mathbb{D}^n \in \mathbb{C}^n$, together with the pseudo-hyperbolic distance and the automorphisms[2], i.e., Möbius self-mappings, is the Poincaré model [Duren and Weir, 2007; Rochberg, 2019]. That is, one can define the unit Poincaré model $\mathbb{P}^n$, with the pseudo-hyperbolic distance, by utilizing the Möbius self-mappings [Duren and Weir, 2007].

When $n = 1$, the Möbius self-mapping on the unit Poincaré disk is defined as the Möbius subtraction in the Möbius gyrovector space [Duren and Weir, 2007]. Specifically, for $z_i, z_j \in \mathbb{P}$, the mapping is given by

$$\varphi_{z_i}(z_j) = z_i \ominus z_j. \tag{4}$$

When $n > 1$, for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{C}^n$ let $P_{\boldsymbol{z}_i}(\boldsymbol{z}_j)$ denote the orthogonal projection of $\boldsymbol{z}_j$ to the subspace $[\boldsymbol{z}_i] = \{\lambda \boldsymbol{z}_i : \lambda \in \mathbb{C}\}$ spanned by $\boldsymbol{z}_i$, and $Q_{\boldsymbol{z}_i}(\boldsymbol{z}_j) = \boldsymbol{z}_j - P_{\boldsymbol{z}_i}(\boldsymbol{z}_j)$ be the projection onto the orthogonal complement of $[\boldsymbol{z}_i]$. We further denote $s_{\boldsymbol{z}_i} = (1 - \|\boldsymbol{z}_i\|^2)^{1/2}$. Then the Möbius self-mapping on $\mathbb{P}^n$ is formulated as

$$\varphi_{\boldsymbol{z}_i}(\boldsymbol{z}_j) = \frac{\boldsymbol{z}_i - P_{\boldsymbol{z}_i}(\boldsymbol{z}_j) - s_{\boldsymbol{z}_i} Q_{\boldsymbol{z}_i}(\boldsymbol{z}_j)}{1 - \langle \boldsymbol{z}_j, \boldsymbol{z}_i \rangle}. \tag{5}$$

Having the Möbius self-mappings at hand, one can define the pseudo-hyperbolic distance in the unit Poincaré ball, given by

$$\rho^n(\boldsymbol{z}_i, \boldsymbol{z}_j) = \|\varphi_{\boldsymbol{z}_i}(\boldsymbol{z}_j)\|, \tag{6}$$

where $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{P}^n$. Then the Eq. (6) can be further transformed to

$$\rho^n(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{1 - \frac{(1 - \|\boldsymbol{z}_i\|^2)(1 - \|\boldsymbol{z}_j\|^2)}{\|1 - \langle \boldsymbol{z}_j, \boldsymbol{z}_i \rangle\|^2}}. \tag{7}$$

Note that the Eq. (7) holds for both $n = 1$ and $n > 1$. In other words, both the pseudo-hyperbolic distance for the unit Poincaré disk and the unit Poincaré ball can be obtained by the Möbius self-mappings.

[2]The automorphism can project the ball to itself. Properties of the automorphism are in the supplementary material.

Drury Arveson Hardy Space

For an n-dimensional open unit ball $\mathbb{D}^n \in \mathbb{C}^n$, a corresponding Drury Arveson Hardy space, denoted by $\mathcal{D}^n$, can be defined as a RKHS on $\mathbb{D}^n$ [Arcozzi *et al.*, 2007]. Specifically, this is realized by a kernel function, given by

$$k^{\mathcal{D}^n}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle k_{\boldsymbol{z}_j}, k_{\boldsymbol{z}_i} \rangle = \frac{1}{1 - \langle \boldsymbol{z}_j, \boldsymbol{z}_i \rangle}, \tag{8}$$

where $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}^n$ and $k_{\boldsymbol{z}} \in \mathcal{D}^n$ is the kernel for the point $\boldsymbol{z}$ in $\mathbb{D}^n$.

Denote $\hat{k}_{\boldsymbol{z}} = \frac{k_{\boldsymbol{z}}}{\|k_{\boldsymbol{z}}\|}$ as the unit vector in the direction of $k_{\boldsymbol{z}}$, one can define the metric for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}^n$ as follows:

$$\delta^{\mathbb{D}^n}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{1 - \|\langle \hat{k}_{\boldsymbol{z}_i}, \hat{k}_{\boldsymbol{z}_j} \rangle\|^2}. \tag{9}$$

Since $k_{\boldsymbol{z}}$ is a vector in $\mathcal{D}^n$, and the right side of Eq. (9) fits the definition of a metric [Bellet *et al.*, 2015], this metric can also be understood as a distance between $k_{\boldsymbol{z}_i}$ and $k_{\boldsymbol{z}_j}$ in $\mathcal{D}^n$.

In this Drury Arveson Hardy space, one can plug the kernel mapping in Eq. (8) into Eq. (9), obtaining the formulation of the DA metric as

$$\delta^{\mathbb{D}^n}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{1 - \frac{(1 - \|\boldsymbol{z}_i\|^2)(1 - \|\boldsymbol{z}_j\|^2)}{\|1 - \langle \boldsymbol{z}_j, \boldsymbol{z}_i \rangle\|^2}}. \tag{10}$$

By observing the Eq. (7) and Eq. (10), we can find that the DA metric $\delta^{\mathbb{D}^n}$ on $\mathbb{D}^n$ equals to the pseudo-hyperbolic distance ρ^n on $\mathbb{P}^n$. In other words, the unit ball $\mathbb{D}^n$ with DA metric $\delta^{\mathbb{D}^n}$ is the Poincaré model $\mathbb{P}^n$ [Rochberg, 2019]. Since $\delta^{\mathbb{D}^n}$ is also a distance metric between $k_{\boldsymbol{z}_i}$ and $k_{\boldsymbol{z}_j}$, the kernel mapping in Eq. (8) can be understood as an isometry from the unit Poincaré model $\mathbb{P}^n$ to the Drury Arveson Hardy space $\mathcal{D}^n$, meaning that the data in the unit Poincaré model $\mathbb{P}^n$ can be embedded into the RKHS without distortion.

3 Isometry from the Curvature-aware Poincaré Model to an RKHS

Section 2 introduces the isometry, realized by a kernel function, which maps the data from the unit Poincaré model to an RKHS. Since the isometric mapping can preserves the distance metric in two space, the distortion issue of embedding the hierarchical data can be addressed. Also, recent studies [Fang *et al.*, 2021; Chami *et al.*, 2019] have verified that the curvature of the hyperbolic model influences the down-stream tasks significantly. This suggests generalizing the isometry to the Poincaré model with adaptive curvature.

In this section, we propose an curvature-aware isometric embedding from the Poincaré model to an RKHS. The embedding is realized by a curvature-aware kernel function. Figure 1 shows the process of this isometric-embedding. Specifically, we first define a pseudo-hyperbolic distance ρ_c^n on the Poincaré model $\mathbb{P}_c^n$, and then construct a new kernel $k_{\boldsymbol{z}}^{\mathcal{D}_c^n}$ on the ball $\mathbb{D}_c^n$. Given the kernel $k_{\boldsymbol{z}}^{\mathcal{D}_c^n}$, a mapping from $\mathbb{D}_c^n$ to the RKHS $\mathcal{D}_c^n$ is realized and a metric $\delta^{\mathbb{D}_c^n}$ on the ball $\mathbb{D}_c^n$ can be defined. Besides, the metric $\delta^{\mathbb{D}_c^n}$ can also be viewed as a distance measure in $\mathcal{D}_c^n$ and we can establish the equivalence between the pseudo-hyperbolic distance ρ_c^n and the metric $\delta^{\mathbb{D}_c^n}$. Therefore, using $\delta^{\mathbb{D}_c^n}$ on $\mathcal{D}_c^n$ as a bridge, an isometry from the Poincaré ball to the RKHS can be established.

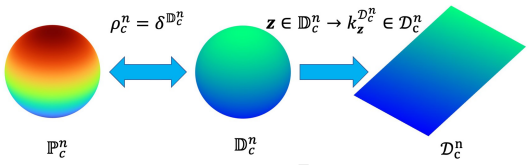

Figure 1: Isometry from the Poincaré model to an RKHS

3.1 Generalization of the Pseudo-hyperbolic Distance in $\mathbb{P}_c^n$

The pseudo-hyperbolic distance in the unit Poincaré model is defined via the norm of Möbius self-mappings (see Eq. (6)) [Duren and Weir, 2007]. To define such a pseudo-hyperbolic distance in the curvature-aware Poincaré ball, one needs to generalize the Möbius self-mapping from $\mathbb{P}^n$ to $\mathbb{P}_c^n$.

Möbius Self-mappings

When $n = 1$, the Möbius self-mappings of the Poincaré disk $\mathbb{P}_c$ is defined as the Möbius subtraction:

$$\varphi_{z_i}^c(z_j) = \frac{z_i - z_j}{1 - c\bar{z}_i z_j}, \tag{11}$$

where $z_i, z_j \in \mathbb{P}_c$ and $\bar{z}_i$ means the complex conjugate of z_i.

We further generalize the Möbius self-mapping from $\mathbb{P}_c$ to $\mathbb{P}_c^n$ and propose the pseudo-hyperbolic distance of $\mathbb{P}_c^n$ in case of $n > 1$. For $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}_c^n$, let $P_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j)$ be the orthogonal projection of $\boldsymbol{z}_j$ onto the subspace $[\boldsymbol{z}_i]$ spanned by $\boldsymbol{z}_i$, and let $Q_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j) = \boldsymbol{z}_j - P_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j)$ be the projection onto the orthogonal complement of $[\boldsymbol{z}_i]$. Therefore, it holds that: $P_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j) = \boldsymbol{0}$ when $\boldsymbol{z}_i = \boldsymbol{0}$, and $P_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j) = \frac{\langle \boldsymbol{z}_j, \boldsymbol{z}_i \rangle}{\langle \boldsymbol{z}_i, \boldsymbol{z}_i \rangle} \boldsymbol{z}_i$ if $\boldsymbol{z}_i \neq \boldsymbol{0}$. Let $s_{\boldsymbol{z}_i}^c = \sqrt{1 - c\boldsymbol{z}_i^2}$, we define the Möbius self-mappings as

$$\varphi_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j) = \frac{\boldsymbol{z}_i - P_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j) - s_{\boldsymbol{z}_i}^c Q_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j)}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i \rangle}, \quad \boldsymbol{z}_i \in \mathbb{D}_c^n. \tag{12}$$

Note that the Eq. (12) can be simplified to $\varphi_{z_i}^c(z_j) = \frac{z_i - z_j}{1 - c\bar{z}_i z_j}$ when $n = 1$.

Here are the properties of the proposed maps $\varphi_{\boldsymbol{z}_i}^c$.

Theorem 1. *For every $\boldsymbol{z}_i \in \mathbb{D}_c^n$, $\varphi_{\boldsymbol{z}_i}^c$ has the following properties:*

- $\varphi_{\boldsymbol{z}_i}^c(\boldsymbol{0}) = \boldsymbol{z}_i$, $\varphi_{\boldsymbol{z}_i}^c(\boldsymbol{z}_i) = \boldsymbol{0}$.
- *$\varphi_{\boldsymbol{z}_i}^c$ is a involution: $\varphi_{\boldsymbol{z}_i}^c(\varphi_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j)) = \boldsymbol{z}_j$.*

Proof. The proof is given in supplementary material. □

According to Theorem 1, $\varphi_{\boldsymbol{z}_i}^c$ is a one-to-one map, and that $\varphi_{\boldsymbol{z}_i}^c = (\varphi_{\boldsymbol{z}_i}^c)^{-1}$. This proves that the Möbius self-mappings are automorphisms.

Pseudo-hyperbolic Distance

With the Möbius self-mappings $\varphi_{\boldsymbol{z}_i}^c$ at our disposal, we can come up with the pseudo-hyperbolic distance of the ball $\mathbb{D}_c^n$, given by

$$\rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{c}\|\varphi_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j)\|, \tag{13}$$

where $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}_c^n$.

In this case, when $n = 1$, the pseudo-hyperbolic distance is defined as

$$\rho_c(z_i, z_j) = \sqrt{c}\Big|\frac{z_i - z_j}{1 - c\bar{z}_j z_j}\Big| = \sqrt{c}d(z_i, z_j), \quad (14)$$

where d is the Möbius gyrodistance (see Eq. (3)) and $|.|$ represents the absolute value of the number. Therefore, the pseudo-hyperbolic distance in Poincaré disk with curvature $-c$ can be rescaled by the Möbius gyrodistance between the gyrovectors. When $c = 1$, it presents to the pseudo-hyperbolic distance in the unit Poincaré disk.

Moreover, when $n > 1$, for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}_c^n$, $\varphi_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j)$ has the following property:

$$\sqrt{c}\|\varphi_{\boldsymbol{z}_i}^c(\boldsymbol{z}_j)\| = \sqrt{1 - \frac{(1 - c\|\boldsymbol{z}_i\|^2)(1 - c\|\boldsymbol{z}_j\|^2)}{\|1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle\|^2}}. \quad (15)$$

It is obvious that when $c = 1$, it becomes the pseudo-hyperbolic distance defined in the unit Poincaré ball. Besides, the Eq. (13) is allowed to predigest to the equation as

$$\rho_c(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{c}\Big\|\frac{\boldsymbol{z}_i - \boldsymbol{z}_j}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle}\Big\|, \quad (16)$$

which can be obtained by rescaling the Möbius gyrodistance between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ in Eq. (3).

Given the analysis, we have the following theorem.

Theorem 2. *The pseudo-hyperbolic distance of the ball $\mathbb{D}_c^n$ has the properties:*

- *$\rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_j) \geq 0$ and $\rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_j) = 0$ if and only if $\boldsymbol{z}_i = \boldsymbol{z}_j$, since $\varphi_{\boldsymbol{z}_i}(\boldsymbol{z}_j) = \mathbf{0}$ only for $\boldsymbol{z}_i = \boldsymbol{z}_j$;*
- *$\rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_j) = \rho_c^n(\boldsymbol{z}_j, \boldsymbol{z}_i)$;*
- *The triangle inequality takes strong form:*

$$\frac{\|\rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_k) - \rho_c^n(\boldsymbol{z}_k, \boldsymbol{z}_j)\|}{1 - \rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_k)\rho_c^n(\boldsymbol{z}_k, \boldsymbol{z}_j)} \leq \rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_j)$$
$$\leq \frac{\rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_k) + \rho_c^n(\boldsymbol{z}_k, \boldsymbol{z}_j)}{1 + \rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_k)\rho_c^n(\boldsymbol{z}_k, \boldsymbol{z}_j)} \quad \textit{for all} \quad \boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{z}_k \in \mathbb{D}_c^n$$

Proof. The proof is given in supplementary material. □

The Theorem 2 shows the pseudo-hyperbolic distance is a true metric of the open ball $\mathbb{D}_c^n$. Furthermore, endowed with the pseudo-hyperbolic distance, the the open ball $\mathbb{D}_c^n$ is the Poincaré model $\mathbb{P}_c^n$.

3.2 RKHS with a Curvature-aware Kernel

In this section, we propose a new kernel function whose induced RKHS has a distance corresponds to the pseudo-hyperbolic distance in Poincaré model.

For $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}_c^n$, we define a curvature-aware kernel in the form of

$$k^{\mathcal{D}_c^n}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle k_{\boldsymbol{z}_j}^{\mathcal{D}_c^n}, k_{\boldsymbol{z}_i}^{\mathcal{D}_c^n}\rangle = \frac{1}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle}, \quad (17)$$

where $\mathbb{D}_c^n = \{\boldsymbol{z} \in \mathbb{C}^n : c\|\boldsymbol{z}\|^2 < 1, c > 0\}$ and $\mathcal{D}_c^n$ is the RKHS with the kernel $k_{\boldsymbol{z}}^{\mathcal{D}_c^n}$. The kernel is PD and the proof is shown in the supplementary material.

After proposing the kernel, we define a metric on the ball similar to Eq. (9) in the form of

$$\delta^{\mathbb{D}_c^n}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{1 - \Big\|\langle \frac{k_{\boldsymbol{z}_j}^{\mathcal{D}_c^n}}{\|k_{\boldsymbol{z}_j}^{\mathcal{D}_c^n}\|}, \frac{k_{\boldsymbol{z}_i}^{\mathcal{D}_c^n}}{\|k_{\boldsymbol{z}_i}^{\mathcal{D}_c^n}\|}\rangle\Big\|^2}. \quad (18)$$

Concretely, it can be written as

$$\delta^{\mathbb{D}_c^n}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{1 - \frac{(1 - c\|\boldsymbol{z}_i\|^2)(1 - c\|\boldsymbol{z}_j\|^2)}{\|1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle\|^2}} \quad (19)$$

by plugging the kernel in Eq. (17) to Eq. (18).

When $n = 1$, the metric (18) can be simplified as

$$\delta^{\mathbb{D}_c}(z_i, z_j) = \sqrt{c}\Big\|\frac{z_i - z_j}{1 - c\bar{z}_i z_j}\Big\|. \quad (20)$$

Through comparing Eq. (20) with Eq. (14), we note $\delta^{\mathbb{D}_c}(z_i, z_j) = \rho_c(z_i, z_j)$. Therefore, the metric is identified with the pseudo-hyperbolic distance of the Poincaré disk with curvature $-c$.

When $n > 1$, compare Eq. (19) with Eq. (15), we know the metric $\delta^{\mathbb{D}_c^n}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ also equals with the pseudo-hyperbolic distance $\rho_c^n(\boldsymbol{z}_i, \boldsymbol{z}_j)$. Therefore, the ball $\mathbb{D}_c^n$ with the metric $\delta^{\mathbb{D}_c^n}(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is a Poincaré model $\mathbb{P}_c^n$.

Furthermore, since (18) can also be viewed as distance in the RKHS $\mathcal{D}_c^n$ by taking $k_{\boldsymbol{z}}^{\mathcal{D}_c^n}$ as a vector in $\mathcal{D}_c^n$, the distance-preserved kernel mapping from the Poincaré model $\mathbb{P}_c^n$ to the RKHS with kernel $k^{\mathcal{D}_c^n}(\boldsymbol{z}_i, \boldsymbol{z}_j) = 1/(1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle)$ realizes an isometry.

4 Curvature-aware Hyperbolic Kernels

In this section, we propose to construct a set of kernels in hyperbolic spaces by leveraging the proposed curvature-aware isometric embedding in Eq. (17).

4.1 Curvature-aware Hyperbolic Linear Kernel

We first come up with the simplest kernel in this context. The embedding in Eq. (17) establishes the isometry from the Poincaré model $\mathbb{P}_c^n$ and the RKHS $\mathcal{D}_c^n$. That said, this embedding can realize the linear kernel in $\mathcal{D}_c^n$, and we term it as curvature-aware hyperbolic linear (CHL) kernel . For $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{P}_c^n$, the CHL kernel is given by

$$k^{\mathrm{CHL}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle k_{\boldsymbol{z}_j}^{\mathcal{D}_c^n}, k_{\boldsymbol{z}_i}^{\mathcal{D}_c^n}\rangle = \frac{1}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle}. \quad (21)$$

4.2 Curvature-aware Hyperbolic Polynomial Kernel

One popular kernel in the Euclidean space is the polynomial (Poly) kernel, which can encode the high-order statistics of the data. For $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{C}^n$, it can be expressed as

$$k^{\mathrm{Poly}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\langle \boldsymbol{x}_j, \boldsymbol{x}_i\rangle + b)^d, \quad b > 0, \quad d > 0. \quad (22)$$

From the formulation to Eq. (22), we can simply extend it in the Poincaré model $\mathbb{P}_c^n$. Specifically, the inner product, which can be understood as the linear kernel in the Euclidean space, in Eq. (22) can be replaced by Eq. (21), thus we can obtain

the curvature-aware hyperbolic polynomial (CHPoly) kernel, formulated by

$$k^{\mathrm{CHPoly}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = (\langle k^{\mathcal{D}_c^n}_{\boldsymbol{z}_j}, k^{\mathcal{D}_c^n}_{\boldsymbol{z}_i}\rangle + b)^d = (\frac{1}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle} + b)^d, \quad b > 0, d > 0. \tag{23}$$

The proposed kernel is a PD one, and we have provided the proof in the supplementary material.

4.3 Curvature-aware Hyperbolic RBF Kernel

In the Euclidean space, the radial basis function (RBF) kernel is recognized as a powerful kernel due to it property of universal approximation [Smola and Schölkopf, 1998]. It takes the form of

$$k^{\mathrm{RBF}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\big(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\tau^2}\big), \tag{24}$$

where $\tau > 0$ is the bandwidth parameter of the RBF kernel. It is difficult to extand the RBF kernel to the Poincaré model since the RBF kernel derived from the geodesic distance in the Poincaré model is not a valid PD kernel [Fang *et al.*, 2021; Jayasumana *et al.*, 2015; Cho *et al.*, 2019]. Fang *et al.* first propose a PD RBF in hyperbolic space in case of using the tangent space. In our work, we extend the RBF kernel in $\mathcal{D}_c^n$, called curvature-aware hyperbolic RBF (CHRBF) kernel. It can be formulated as

$$k^{\mathrm{CHRBF}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\frac{\|k^{\mathcal{D}_c^n}_{\boldsymbol{z}_i} - k^{\mathcal{D}_c^n}_{\boldsymbol{z}_j}\|^2}{2\tau^2}\big), \tag{25}$$

for $\tau > 0$. It is worth noting that

$$\begin{aligned}&\|k^{\mathcal{D}_c^n}_{\boldsymbol{z}_i} - k^{\mathcal{D}_c^n}_{\boldsymbol{z}_j}\|^2\\ =&k^{\mathrm{CHL}}(\boldsymbol{z}_i, \boldsymbol{z}_i) + k^{\mathrm{CHL}}(\boldsymbol{z}_j, \boldsymbol{z}_j) - 2k^{\mathrm{CHL}}(\boldsymbol{z}_i, \boldsymbol{z}_j).\end{aligned} \tag{26}$$

Plug Eq. (26) into Eq. (25), we can obtain the full fumulation of CHRBF kernel as

$$k^{\mathrm{CHRBF}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\Big(-\frac{1}{2\tau^2}\big(\frac{1}{1 - c\langle \boldsymbol{z}_i, \boldsymbol{z}_i\rangle} + \frac{1}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_j\rangle} - \frac{2}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle}\big)\Big). \tag{27}$$

Of note, in our formulation, the CHRBF kernel is also a valid PD kernel.

4.4 Curvature-aware Hyperbolic Laplacian Kernel

Another famous exponential-type kernel in the Euclidean space is known as the Laplacian (Lap) kernel. Its formulation is given by

$$k^{\mathrm{Lap}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\big(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\tau}\big), \quad \tau > 0. \tag{28}$$

Analogously, we can deduce the curvature-aware hyperbolic Laplacian kernel (CHLap) kernel as

$$k^{\mathrm{CHLap}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\frac{\|k^{\mathcal{D}_c^n}_{\boldsymbol{z}_i} - k^{\mathcal{D}_c^n}_{\boldsymbol{z}_j}\|}{\tau}\big), \quad \tau > 0. \tag{29}$$

Following the process of develop the CHRBF kernel in Eq. (27), one can yield the formulation of CHLap kernel as:

$$k^{\mathrm{CHLap}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\Big(-\frac{1}{\tau}\big(\frac{1}{1 - c\langle \boldsymbol{z}_i, \boldsymbol{z}_i\rangle} + \frac{1}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_j\rangle} - \frac{2}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle}\big)^{\frac{1}{2}}\Big). \tag{30}$$

The PD property of CHLap kernel is also proven in the supplementary material.

4.5 Curvature-aware Hyperbolic Sigmoid Kernel

The former proposed kernels are PD. As a complementary concept of the PD kernel, the indefinite kernel has also been studied in the learning community [Ong *et al.*, 2004]. In contrast to PD kernels, the indefinite kernel maps the data to an RKKS with indefinite inner product [Ong *et al.*, 2004]. In [Cho *et al.*, 2019], an indefinite kernel has been investigated and shows its effectiveness. This inspires us to further develop an indefinite kernel using the proposed embedding(Eq. (17) or Eq. (21)). In this case, we leverage the Sigmoid (Sig) kernel to develop an indefinite kernel. For $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{C}^n$, the Sig kernel can be expressed as

$$k^{\mathrm{Sig}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\big(\gamma\langle \boldsymbol{x}_j, \boldsymbol{x}_i\rangle + \theta\big), \tag{31}$$

where $\gamma > 0$ and $\theta < 0$. This is an indefinite kernel [Bishop and Nasrabadi, 2006]. Plug the Eq. 21 into Eq. 31, we can obtain the curvature-aware hyperbolic Sigmoid (CHSig) kernel for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{P}_c^n$, as

$$k^{\mathrm{CHSig}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \tanh\Big(\gamma\langle k^{\mathcal{D}_c^n}_{\boldsymbol{z}_i}, k^{\mathcal{D}_c^n}_{\boldsymbol{z}_j}\rangle + \theta\Big) = \tanh\Big(\frac{\gamma}{1 - c\langle \boldsymbol{z}_j, \boldsymbol{z}_i\rangle} + \theta\Big), \tag{32}$$

where $\gamma > 0$ and $\theta < 0$. Since the Sig kernel is an indefinite kernel, the proposed CHSig kernel is also an indefinite kernel, which maps the hyperbolic data from the Poincaré model to an RKKS. The optimization algorithms in RKKSs can be used to address the tasks in hyperbolic space [Loosli *et al.*, 2015; Xu *et al.*, 2017].

5 Experiments

In this section, we conduct thorough experiments to evaluate the superiority of the proposed kernels. Source code and Appendix are available at https://github.com/MMeiYang/Code-and-Appendix-for-Expanding-the-Hyperbolic-Kernels-A-Curvature-aware-Isometric-Embedding-View.

5.1 Graph Learning

Learning on the graphs has gained significant attention in the learning community since the graph-structured data are popular to model the message passing networks including social networks, citation networks and communication networks [Xia *et al.*, 2021]. We first evaluate the proposed kernels over the graph data with existing hyperbolic kernels and Euclidean kernels. The hyperbolic kernels include hyperbolic polynomial (HPoly) kernel proposed by [Cho *et al.*, 2019], hyperbolic tangent (HTang) kernel, hyperbolic RBF

Dataset \ Kernel	Dim	Hyperbolic Embedding										Euclidean Embedding				
		CHL	CHPoly	CHRBF	CHLap	CHSig	HTang	HPoly	HRBF	HLap	HBin	EL	EPoly	ERBF	ELap	ESig
Facebook	#2	$84.3_{1.0}$	$82.0_{1.8}$	$77.5_{0.9}$	$80.1_{0.7}$	$\mathbf{86.5}_{0.5}$	$63.3_{0.6}$	$66.6_{2.8}$	$74.3_{1.0}$	$74.8_{1.2}$	$67.5_{1.3}$	$61.4_{0.2}$	$66.0_{0.5}$	$67.0_{0.4}$	$67.3_{0.3}$	$70.7_{0.5}$
	#5	$89.9_{0.3}$	$89.7_{0.4}$	$88.0_{0.8}$	$88.1_{0.7}$	$\mathbf{90.7}_{0.4}$	$65.2_{0.5}$	$81.6_{1.1}$	$87.2_{0.8}$	$87.9_{0.6}$	$86.0_{1.2}$	$64.1_{0.6}$	$82.5_{0.8}$	$84.7_{0.6}$	$85.6_{0.7}$	$86.5_{0.7}$
	#10	$90.6_{0.4}$	$90.6_{0.3}$	$88.4_{0.6}$	$88.8_{0.3}$	$\mathbf{90.8}_{0.4}$	$74.0_{0.8}$	$86.5_{1.1}$	$88.2_{0.7}$	$88.5_{0.5}$	$88.4_{1.4}$	$75.8_{0.9}$	$86.4_{0.4}$	$86.8_{0.8}$	$86.7_{0.6}$	$88.9_{0.4}$
	#25	$90.8_{0.2}$	$90.9_{0.3}$	$88.4_{0.6}$	$88.6_{0.4}$	$\mathbf{91.3}_{0.4}$	$85.8_{0.5}$	$88.9_{0.6}$	$88.0_{0.5}$	$88.2_{0.7}$	$89.6_{1.0}$	$83.7_{0.3}$	$88.5_{0.6}$	$87.9_{0.6}$	$88.1_{0.6}$	$89.4_{0.3}$
Terrorist	#2	$60.5_{3.6}$	$57.6_{2.2}$	$63.3_{1.2}$	$\mathbf{64.1}_{1.1}$	$59.9_{0.8}$	$49.8_{1.9}$	$51.3_{0.7}$	$60.5_{1.6}$	$62.6_{2.0}$	$55.0_{3.0}$	$50.0_{1.8}$	$51.3_{0.8}$	$60.0_{1.6}$	$60.9_{2.1}$	$61.0_{1.3}$
	#5	$68.1_{1.1}$	$\mathbf{68.4}_{1.3}$	$67.5_{2.1}$	$67.9_{2.1}$	$67.6_{1.9}$	$49.5_{1.8}$	$52.8_{1.3}$	$66.3_{2.1}$	$67.2_{2.0}$	$67.4_{2.1}$	$52.5_{0.7}$	$56.2_{1.3}$	$62.1_{2.3}$	$63.8_{1.5}$	$61.5_{1.4}$
	#10	$\mathbf{69.8}_{2.2}$	$69.4_{1.9}$	$68.2_{1.7}$	$68.5_{2.4}$	$67.7_{2.3}$	$51.6_{1.5}$	$59.3_{1.6}$	$67.0_{1.9}$	$66.8_{3.5}$	$68.3_{2.1}$	$58.7_{1.7}$	$65.4_{2.0}$	$65.0_{1.3}$	$66.6_{1.8}$	$64.9_{1.5}$
	#25	$68.6_{2.8}$	$\mathbf{69.1}_{2.0}$	$66.9_{1.3}$	$68.0_{1.4}$	$67.8_{1.9}$	$53.3_{1.4}$	$60.8_{1.7}$	$66.4_{1.9}$	$67.0_{1.2}$	$67.0_{1.5}$	$60.3_{2.1}$	$66.1_{1.5}$	$62.1_{2.7}$	$63.2_{2.3}$	$65.2_{1.9}$
Wiki	#2	$\mathbf{58.7}_{1.5}$	$50.5_{2.8}$	$47.6_{1.1}$	$49.8_{1.8}$	$58.4_{1.1}$	$18.7_{0.8}$	$17.0_{2.5}$	$41.4_{2.3}$	$43.1_{1.6}$	$33.0_{1.7}$	$18.2_{0.9}$	$18.4_{1.0}$	$23.7_{1.0}$	$24.4_{1.1}$	$25.5_{0.9}$
	#5	$\mathbf{71.1}_{0.6}$	$69.4_{1.1}$	$66.4_{0.8}$	$67.1_{1.1}$	$70.9_{0.8}$	$30.5_{3.9}$	$58.4_{1.2}$	$65.1_{1.1}$	$66.3_{1.6}$	$60.5_{1.3}$	$26.6_{2.6}$	$45.5_{1.8}$	$57.6_{1.4}$	$58.5_{1.4}$	$62.2_{0.6}$
	#10	$\mathbf{73.2}_{0.6}$	$72.3_{0.6}$	$68.3_{1.6}$	$69.1_{1.6}$	$71.8_{1.1}$	$38.4_{3.4}$	$62.0_{0.4}$	$67.4_{0.9}$	$68.2_{1.4}$	$64.7_{2.1}$	$42.7_{1.9}$	$57.8_{2.3}$	$65.6_{1.5}$	$65.2_{1.5}$	$67.8_{0.8}$
	#25	$\mathbf{73.6}_{1.0}$	$73.1_{0.5}$	$69.7_{0.9}$	$71.0_{0.9}$	$72.1_{0.6}$	$59.9_{0.7}$	$67.5_{1.6}$	$69.7_{1.0}$	$69.8_{1.3}$	$66.1_{0.8}$	$58.4_{1.8}$	$64.6_{1.5}$	$68.6_{1.5}$	$68.1_{1.3}$	$70.7_{0.8}$
AC	#2	$76.3_{0.9}$	$75.4_{1.1}$	$75.7_{0.8}$	$77.3_{0.8}$	$\mathbf{78.8}_{0.7}$	$66.0_{1.2}$	$61.4_{7.0}$	$74.8_{1.3}$	$75.9_{0.8}$	$72.2_{1.5}$	$58.6_{4.0}$	$64.9_{4.8}$	$72.3_{1.6}$	$72.3_{1.3}$	$64.9_{4.8}$
	#5	$86.3_{0.8}$	$85.9_{0.8}$	$85.5_{0.8}$	$85.1_{1.1}$	$\mathbf{86.7}_{1.0}$	$75.9_{1.4}$	$79.1_{0.8}$	$85.1_{1.0}$	$85.1_{1.4}$	$84.1_{0.9}$	$68.4_{2.5}$	$71.6_{3.3}$	$79.0_{0.8}$	$79.2_{0.6}$	$71.6_{3.3}$
	#10	$\mathbf{89.5}_{0.6}$	$88.8_{0.5}$	$87.6_{0.8}$	$88.1_{0.8}$	$89.3_{0.6}$	$82.0_{0.6}$	$86.7_{1.1}$	$87.6_{0.7}$	$87.6_{0.5}$	$87.1_{1.2}$	$79.4_{0.4}$	$80.0_{3.2}$	$85.0_{0.8}$	$84.4_{0.9}$	$80.0_{3.2}$
	#25	$89.6_{0.5}$	$89.3_{0.4}$	$87.5_{0.5}$	$87.8_{0.7}$	$\mathbf{89.7}_{0.4}$	$85.4_{0.4}$	$87.7_{0.8}$	$87.1_{0.7}$	$87.0_{0.7}$	$87.5_{0.7}$	$81.6_{0.8}$	$83.4_{1.2}$	$86.8_{0.7}$	$86.8_{0.6}$	$83.4_{1.2}$
Cora ML	#2	$69.0_{0.5}$	$68.5_{0.9}$	$68.5_{1.0}$	$69.2_{0.8}$	$\mathbf{70.3}_{0.6}$	$38.2_{1.6}$	$47.8_{0.4}$	$67.9_{0.6}$	$69.0_{0.7}$	$58.7_{1.3}$	$39.7_{3.4}$	$45.9_{2.0}$	$57.0_{0.7}$	$57.7_{0.6}$	$60.4_{0.4}$
	#5	$83.6_{0.5}$	$84.0_{0.8}$	$82.8_{0.8}$	$83.6_{0.5}$	$\mathbf{84.1}_{0.6}$	$62.7_{1.5}$	$72.5_{1.7}$	$82.7_{0.9}$	$83.3_{0.5}$	$80.7_{1.1}$	$45.1_{1.7}$	$58.2_{0.9}$	$70.7_{0.8}$	$71.5_{0.6}$	$73.4_{0.6}$
	#10	$86.0_{0.2}$	$\mathbf{85.8}_{0.7}$	$84.6_{0.3}$	$85.1_{0.4}$	$\mathbf{85.8}_{0.5}$	$72.3_{0.9}$	$80.8_{0.6}$	$84.3_{0.6}$	$85.0_{0.6}$	$83.7_{0.5}$	$62.6_{0.8}$	$71.5_{1.1}$	$76.2_{0.8}$	$77.2_{0.8}$	$77.1_{0.4}$
	#25	$86.1_{0.2}$	$\mathbf{86.2}_{0.6}$	$84.8_{0.8}$	$85.6_{0.6}$	$86.0_{0.5}$	$79.2_{0.4}$	$83.5_{0.9}$	$84.9_{0.4}$	$85.2_{0.7}$	$84.9_{0.6}$	$72.0_{0.7}$	$75.9_{1.3}$	$79.0_{0.7}$	$78.4_{0.5}$	$79.0_{0.6}$
Avg. ACC.		**78.3**	77.3	75.7	76.6	**78.3**	60.1	67.6	74.8	75.4	72.6	58.0	65.0	69.9	70.3	70.2
Top1 Times		6	4	0	1	10	0	0	0	0	0	0	0	0	0	0

Table 1: Mean accuracy (%) of node classification on graph datasets including Facebook, Terrorist, Wiki, AC and Cora ML. The subscript of each number indicates the corresponding standard deviation. We use **bold** to indicate the best result.

(HRBF) kernel, hyperbolic Laplace (HLap) kernel and hyperbolic Binomial (HBin) kernel in [Fang *et al.*, 2021] and the Euclidean kernels includes Euclidean linear (EL) kernel, Euclidean polynomial (EPoly) kernel, Euclidean RBF (ERBF) kernel, Euclidean Laplacian (ELap) kernel and Euclidean Sigmoid (ESig) kernel.

Specifically, the graph are first mapped into the Poincaré model for hyperbolic kernels, while the graph are mapped into the Euclidean space for Euclidean kernels, and then the graph nodes are classified by kernel SVMs. The details of the SVM solvers for PD and indefinite kernels are in supplementary material.

Datasets and Evaluation Protocol

Five real-world graph datasets including Facebook [Rozemberczki *et al.*, 2019], Terrorist [Zhao *et al.*, 2006], Wiki [Cucerzan, 2007], Amazon Electronics Computers (AC) [Shchur *et al.*, 2018], Cora ML [Bojchevski and Günnemann, 2017] are used in this study. Both the hyperbolic space and the Euclidean space are employed as the embedding space with 2, 5, 10, 25 dimensions. The details of the datasets are provided in the supplementary material.

In addition, an one-vs-all (OVA) strategy is utilized to implement multi-class classification and Platt scaling maps the predicted scores on the hold-out data to probabilities for all categories [Platt and others, 1999]. We use the mean classification accuracy (ACC) to measure the performance of all the methods. Due to space limitation, other two performance metrics including area under the ROC curve (AUC) and macro-averaged area under precision recall curve (AUPR) are reported in the supplementary material.

Experimental Results

We first compare our proposed hyperbolic kernel methods with popular Euclidean kernel methods. The results are shown in Table 1. We observe that each of our hyperbolic kernels shows a great improvement against its Euclidean counterpart. For example, our CHL, CHPoly, CHRBF, CHLap and CHSig kernels improve the mean ACC by 20.3%, 12.3%, 5.8%, 6.3% and 8.1% compared with the Euclidean counterparts including EL, EPloy, ERBF, ELap and ESig kernels respectively. This shows that the hyperbolic space is a better option to encode the graph-structured data, and can improve the representation power of the graphs.

Next, the state-of-the-art hyperbolic kernel methods are employed as baseline to further demonstrate the superiority of our proposed kernels. Table 1 illustrates that our methods also obviously improve classification performance compared with existing hyperbolic kernel methods. For example, even the CHRBF kernel with the lowest performance in our proposed hyperbolic kernels improves the ACC value over HTang, HPoly, HRBF, HLap and HBin kernels by 15.6%, 8.1%, 0.9%, 0.3%, 3.1%, respectively. Compared with the HPoly kernel which fixes the curvature of hyperbolic space as -1, the good performance of our proposed kernels on different datasets demonstrates that curvature-aware hyperbolic kernels can improve the adaptability. In addition, comparisons with HTang, HRBF, HLap and HBin kernels further demonstrate the superiority of our method, which shows that the isometry from Poincaré model to the RKHS reduces the distortion of mapping hyperbolic data to Euclidean space.

5.2 Zero-shot Learning

Zero-shot Learning (ZSL) portrays the problem of recognizing the unseen object by matching its semantic features [Akata *et al.*, 2016; Xian *et al.*, 2018], and can be modeled as a cross-modality matching task. We follow the practice in [Fang *et al.*, 2021] to train the network and the paradigm is shown in Figure 2. For a batch of images, e.g., $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N]$, and all semantic embedding, e.g., $\boldsymbol{E} = [\boldsymbol{E}_1, \ldots, \boldsymbol{E}_M]$, as input, the visual features (e.g., $\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$) and semantic features (e.g., $\boldsymbol{e} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_M]$) are extracted by two neural networks, as $\boldsymbol{x} = f(\boldsymbol{X})$ and $\boldsymbol{e} = g(\boldsymbol{E})$. Having two-modality of data at hand, the network can be optimized by the alignment loss.

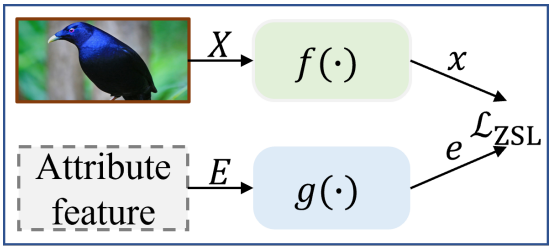

Figure 2: The learning paradigm of the zero-shot learning task.

In the experiments, we follow the loss function of baseline in [Fang *et al.*, 2021] to optimize the network. More details about the loss functions for ZSL are demonstrated in supplementary material.

Datasets and Evaluation Protocol

We use CUB [Wah *et al.*, 2011], AWA1 [Lampert *et al.*, 2013] and AWA2 [Akata *et al.*, 2016] to evaluate the ZSL task. The statistics of the dataset is provided in the supplemental material. The model performance is calculated by the harmonic mean (HM) score, i.e., $\mathrm{HM} = \frac{2\times \mathrm{S}\times \mathrm{U}}{\mathrm{U}+\mathrm{S}}$, where S and U denote the mean accuracy of seen and unseen classes respectively.

Model	CUB			AWA1			AWA2		
	U	S	HM	U	S	HM	U	S	HM
LATEM [Xian *et al.*, 2016]	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0
DEVISE [Frome *et al.*, 2013]	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8
DEM [Zhang *et al.*, 2017]	19.6	57.9	29.2	32.8	84.7	47.3	30.5	86.4	45.1
ALE [Akata *et al.*, 2016]	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9
SP-AEN [Chen *et al.*, 2018]	34.7	**70.6**	46.6	-	-	-	23.3	**90.9**	37.1
CRnet [Zhang and Shi, 2019]	45.5	56.8	50.5	58.1	74.7	65.4	52.6	78.8	63.1
Kai et al. [Li *et al.*, 2019]	**47.4**	47.6	47.5	**62.7**	77.0	69.1	56.4	81.4	66.7
HTang kernel [Fang *et al.*, 2021]	40.8	58.1	47.9	52.3	85.2	64.8	37.1	88.5	52.3
HRBF kernel [Fang *et al.*, 2021]	44.6	57.8	50.3	59.0	84.6	69.5	42.9	89.5	57.9
HLap kernel [Fang *et al.*, 2021]	46.2	56.1	50.7	60.7	83.5	70.3	54.1	87.1	66.7
HBin kernel [Fang *et al.*, 2021]	39.8	56.9	46.8	43.7	**88.9**	58.6	39.8	90.5	55.4
Baseline [Fang *et al.*, 2021]	18.6	44.6	26.3	29.8	76.4	42.9	25.5	76.4	38.2
CHL kernel	43.3	58.3	49.7	51.2	84.7	63.8	44.5	90.8	59.8
CHPoly kernel	41.7	58.9	48.8	51.3	85.4	64.1	42.2	**90.9**	57.6
CHRBF kernel	45.0	56.7	50.1	56.3	82.7	67.0	45.1	90.1	60.1
CHLap kernel	45.2	56.1	50.1	53.4	**88.9**	66.7	44.9	**90.9**	60.1
CHSig kernel	46.9	56.5	**51.3**	59.3	87.2	**70.6**	**69.2**	73.5	**71.3**

Table 2: Zero-shot recognition results (%) on the CUB, AWA1, AWA2 datasets. The harmonic mean (HM) of classification accuracy of unseen classes (U) and seen classes (S) are reported. We use **bold** to indicate the best result.

Experimental Results

We first evaluate our methods by comparing them with the baseline. As shown in Table 2, all our kernels show obvious improvements to the baseline. Concretely, compared with the baseline, even proposed simplest CHL kernel improve 23.4%, 20.9% and 21.6% on HM for CUB, AWA1 and AWA2. Besides, the powerful indefinite CHSig kernel further improves the accuracy of ZSL, which demonstrates the superiority of hyperbolic kernel for representation learning.

In addition, to further show the superiority of our method, a series of ZSL methods are applied for comparison, including the non-generative algorithms [Zhang and Shi, 2019; Li *et al.*, 2019]. We note that our proposed CHSig kernel yields the best results on all datasets.

Finally, we compare our methods with the state-of-the-art methods with hyperbolic kernels for ZSL [Fang *et al.*, 2021]. We observe from Table 2 that our methods are still very competitive compared with the existing hyperbolic kernel methods. In particular, our CHSig kernel outperforms existing methods across all 3 datasets, clearly showing the superiority of the proposed kernels.

5.3 Further Studies

In this part, we study the effect of the curvature of the Poincaré ball. We conduct this study in the graph learning task. Specifically, the Facebook dataset and five kernels, e.g., CHL kernel, CHPoly kernel, CHRBF kernel, CHLap kernel, and CHSig kernel, are used in this analysis. In the experiments, we set c to $\{2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}\}$ while the curvature value is represented as $-c$. Figure 3 shows the results. We can observe that: (1) In the regime of the small c value, e.g., $c < 0.5$, the overall trend of classification accuracy of the all kernels is leveled off. (2) However, for PD kernels, it also shows that a large value of the c will degrade the classification performance. (3) The indefinite kernel, i.e., is insensitive to the curvature, indicating its superiority and robustness. This study indeed shows the discrimination of the Poincaré model can benefit from the curvature.

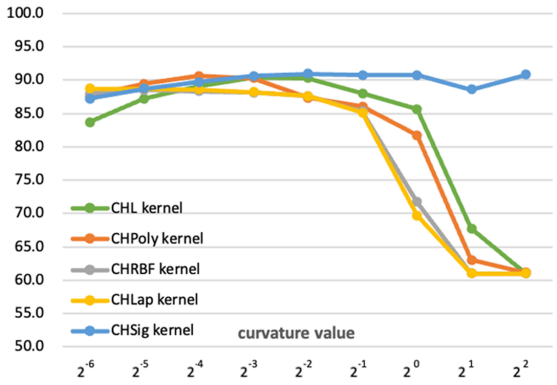

Figure 3: The trend of classification performance of curvature-aware hyperbolic kernels with curvature variation. The experiments are implemented on the Facebook embedded in 10-dimensional Poincaré model. The coordinates on the horizontal axis represent the value of c and the coordinates on the vertical axis represent the value of ACC.

6 Conclusion

This paper proposed a family of hyperbolic kernels with a new curvature-aware isometric embedding function. This embedding was realized by a kernel mapping, which projected the hyperbolic space to an RKHS. Then we developed a set of kernels including several PD kernels and an indefinite kernel. The proposed kernels can adapt to different data regimes and preserve the raw data's hierarchical structure. Extensive experiments on graph learning and zero-shot learning evaluate the effectiveness of the kernels. We believe the proposed isometric-embedding function can construct more powerful kernel machines.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62076062) and the Social Development Science and Technology Project of Jiangsu Province (No.BE2022811). Furthermore, the work was also supported by the Collaborative Innovation Center of Wireless Communications Technology and the Big Data Computing Center of Southeast University.

References

[Akata *et al.*, 2016] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, July 2016.

[Arcozzi *et al.*, 2007] Nicola Arcozzi, Richard Rochberg, and Eric Sawyer. The diameter space, a restriction of the drury-arveson-hardy space. In *Proceedings of the Conference on Function Spaces*, pages 21–42. Providence, R.I. : American Mathematical Society, January 2007.

[Bellet *et al.*, 2015] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 9(1):1–151, 2015.

[Bishop and Nasrabadi, 2006] Christopher M Bishop and Nasser M Nasrabadi. *Pattern Recognition and Machine Learning*. Number 4. Springer, 2006.

[Bojchevski and Günnemann, 2017] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

[Cannon *et al.*, 1997] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of Geometry*, 31:59–115, 1997.

[Chami *et al.*, 2019] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 4869–4880, 2019.

[Chen *et al.*, 2018] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1043–1052, 2018.

[Cho *et al.*, 2019] Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. Large-margin classification in hyperbolic space. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 1832–1840, 2019.

[Coxeter and Greitzer, 1967] Harold Scott Macdonald Coxeter and Samuel L Greitzer. *Geometry revisited*, volume 19. Maa, 1967.

[Cucerzan, 2007] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708–716, 2007.

[Duren and Weir, 2007] Peter Duren and Rachel Weir. The pseudohyperbolic metric and bergman spaces in the ball. *Transactions of the American Mathematical Society*, 359(1):63–76, 2007.

[Fang *et al.*, 2021] Pengfei Fang, Mehrtash Harandi, and Lars Petersson. Kernel methods in hyperbolic spaces. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10665–10674, October 2021.

[Fang *et al.*, 2023] Pengfei Fang, Mehrtash Harandi, Trung Le, and Dinh Phung. Hyperbolic geometry in computer vision: A survey. *arXiv preprint arXiv:2304.10764*, 2023.

[Frome *et al.*, 2013] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[Ganea *et al.*, 2018a] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the International Conference on Machine Learning*, pages 1632–1641, 2018.

[Ganea *et al.*, 2018b] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 31, pages 5345–5355, 2018.

[Gulcehre *et al.*, 2018] 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. *arXiv preprint arXiv:1805.09786*, 2018.

[Jayasumana *et al.*, 2015] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on riemannian manifolds with gaussian RBF kernels. *PAMI*, 2015.

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

[Lampert *et al.*, 2013] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2013.

[Li *et al.*, 2019] Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3583–3592, 2019.

[Loosli *et al.*, 2015] Gaëlle Loosli, Stéphane Canu, and Cheng Soon Ong. Learning SVM in Kreĭn spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1204–1216, 2015.

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

[Ong *et al.*, 2004] Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander J Smola. Learning with non-positive kernels. In *Proceedings of the International Conference on Machine Learning*, page 81. ACM, 2004.

[Platt and others, 1999] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

[Ratcliffe, 1994] John G Ratcliffe. *Foundations of Hyperbolic Manifolds*. Springer, 1994.

[Rochberg, 2019] Richard Rochberg. Complex hyperbolic geometry and hilbert spaces with complete pick kernels. *Journal of Functional Analysis*, 276(5):1622 – 1679, 2019.

[Rozemberczki *et al.*, 2019] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. Gemsec: Graph embedding with self clustering. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 65–72. ACM, 2019.

[Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[Shimizu *et al.*, 2020] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020.

[Smola and Schölkopf, 1998] Alex J Smola and Bernhard Schölkopf. *Learning with kernels*, volume 4. Citeseer, 1998.

[Ungar, 1998] Abraham A Ungar. From pythagoras to einstein: the hyperbolic pythagorean theorem. *Foundations of physics*, 28(8):1283–1321, 1998.

[Ungar, 2022] Abraham Albert Ungar. Analytic hyperbolic geometry and albert einstein's special theory of relativity , world sci. *Publ. Co., Singapore*, 2022.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2010-001, California Institute of Technology, 2011.

[Xia *et al.*, 2021] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021.

[Xian *et al.*, 2016] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

[Xian *et al.*, 2018] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning–a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018.

[Xu *et al.*, 2017] Haiming Xu, Hui Xue, Xiaohong Chen, and Yunyun Wang. Solving indefinite kernel support vector machine with difference of convex functions programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[Zhang and Shi, 2019] Fei Zhang and Guangming Shi. Co-representation network for generalized zero-shot learning. In *Proceedings of the International Conference on Machine Learning*, pages 7434–7443. PMLR, 2019.

[Zhang *et al.*, 2017] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2030, 2017.

[Zhao *et al.*, 2006] Bin Zhao, Prithviraj Sen, and Lise Getoor. Entity and elationship labeling in affiliation networks. In *ICML Workshop on Statistical Network Analysis*, 2006.