Kernel Methods in Hyperbolic Spaces

Pengfei Fang[1,3], Mehrtash Harandi[2,3], Lars Petersson[3]
[1]The Australian National University, [2]Monash University, [3]DATA61-CSIRO
Pengfei.Fang@anu.edu.au, mehrtash.harandi@monash.edu, Lars.Petersson@data61.csiro.au

Abstract

*Embedding data in hyperbolic spaces has proven beneficial for many advanced machine learning applications such as image classification and word embeddings. However, working in hyperbolic spaces is not without difficulties as a result of its curved geometry (*e.g.*, computing the Frechet mean of a set of points requires an iterative algorithm). Furthermore, in Euclidean spaces, one can resort to kernel machines that not only enjoy rich theoretical properties but that can also lead to superior representational power (*e.g.*, infinite-width neural networks). In this paper, we introduce positive definite kernel functions for hyperbolic spaces. This brings in two major advantages, **1.** kernelization will pave the way to seamlessly benefit from kernel machines in conjunction with hyperbolic embeddings, and **2.** the rich structure of the Hilbert spaces associated with kernel machines enables us to simplify various operations involving hyperbolic data. That said, identifying valid kernel functions on curved spaces is not straightforward and is indeed considered an open problem in the learning community. Our work addresses this gap and develops several valid positive definite kernels in hyperbolic spaces, including the universal ones (*e.g.*, RBF). We comprehensively study the proposed kernels on a variety of challenging tasks including few-shot learning, zero-shot learning, person re-identification and knowledge distillation, showing the superiority of the kernelization for hyperbolic representations.*

1. Introduction

This paper proposes a family of positive definite (pd) kernels to map the representations in hyperbolic spaces into Reproducing Kernel Hilbert Spaces (RKHSs), which enables us to seamlessly benefit from kernel machines to analyze hyperbolic spaces.

In the machine learning community, the Euclidean space has been the "workhorse" for feature embeddings. This is mainly because the high-dimensional vector space is a natural generalization from the familiar three-dimensional space we live in and performing basic operations for comparison (*e.g.*, calculating distances and similarities) is straightforward. However, embedding in Euclidean spaces can harm and distort the encoding of structured data, thereby losing the complex geometric information inherently present in the data. For example, the Euclidean space fails to encode the hierarchical information in graph-structured data [38].

Several recent studies in computer vision suggest that embedding images and video using hyperbolic geometry can be beneficial compared to the common practice of using Euclidean geometry. This includes tasks such as textual entailment [18], image classification and retrieval [32], and graph classification [38] to name a few.

The hyperbolic space is characterized by a constant negative sectional curvature (in contrast to the flat structure of the Euclidean space), and does not satisfy Euclid's parallel postulate. One intriguing property of hyperbolic spaces is their capacity of encoding hierarchical data, as the volume of hyperbolic space expands exponentially [22], thereby increasing their representation power. Although several studies have successfully employed the hyperbolic geometry for inference [18, 32, 8], the difficulties of working with such non-linear spaces still overwhelm their wider use. For example, while averaging in Euclidean geometry is straightforward, its counterpart in hyperbolic space is approximated by the Frechet mean. Computing the Frechet mean requires an iterative algorithm and could easily become costly [31, 40]. This motivates us to develop kernels to make it possible to seamlessly benefit and employ kernel machines towards analyzing hyperbolic data.

To be able to make use of kernel machines, one needs to have a pd kernel function at its disposal. Loosely speaking, a kernel function is a measure of similarity. Many familiar kernels in the Euclidean space are defined as functions of the Euclidean distance (which is indeed the geodesic distance of the space). Take the RBF kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\xi d^2(\boldsymbol{x}, \boldsymbol{y}))$ as an example. This might imply that valid pd kernels in curved spaces, the hyperbolic space being one, can be constructed once the geodesic distance is known. Unfortunately, this is not the case as shown in [30, 15] (*c.f.*, theorem 6.2 in [30]), because such curved

Table 1. Summary of the proposed positive definite kernels in hyperbolic spaces and their properties.

Kernel	Formulation: $k(\boldsymbol{z}_i, \boldsymbol{z}_j)$	Condition	Properties
$f_{\mathbb{D}}(\boldsymbol{z}) = \tanh^{-1}(\sqrt{c}\\|\boldsymbol{z}\\|)\frac{\boldsymbol{z}}{\sqrt{c}\\|\boldsymbol{z}\\|},\ c > 0$ and $\boldsymbol{z} \in \mathbb{D}_c^n$			
Hyperbolic tangent kernel	$k^{\tan}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j)\rangle$	-	pd
Hyperbolic RBF kernel	$k^{\mathrm{rbf}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\xi\\|f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j)\\|^2\big)$	$\xi > 0$	pd, universal
Hyperbolic Laplace kernel	$k^{\mathrm{lap}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\xi\\|f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j)\\|\big)$	$\xi > 0$	pd, universal
Generalized Hyperbolic Laplace kernel	$k^{\mathrm{glap}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\xi\\|f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j)\\|^{2\alpha}\big)$	$\xi > 0, 0 < \alpha < 1$	pd, universal
Hyperbolic binomial kernel	$k^{\mathrm{bin}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \big(1 - \langle f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j)\rangle\big)^{-\alpha}$	$\alpha > 0$	pd, universal

spaces are not isometric to flat Euclidean spaces. Interestingly, the difficulty of defining pd kernels on curved spaces is now considered an open problem in machine learning [14].

In this paper, we address the design challenge of pd kernels for hyperbolic representations using the Poincaré model. Here, we propose several valid pd hyperbolic kernels, including the powerful universal ones. To this end, we first make use of a lemma to construct a valid linear-like kernel. Leveraging this lemma, we further define valid RBF and Laplace kernels for the hyperbolic geometry. Finally, we propose the binomial kernel. Table 1 summarizes the proposed kernels. The **contributions** of this work include:

- We propose four pd kernels for the hyperbolic spaces, namely, the hyperbolic tangent kernel, the hyperbolic RBF kernel, the hyperbolic Laplace and the hyperbolic binomial kernel, in conjunction with their theoretical analysis. To the best of our knowledge, this is the first work to develop pd kernels in hyperbolic spaces.

- To evaluate the power of the proposed kernels, we conduct thorough experiments on various vision tasks including few-shot learning, zero-shot learning, person re-identification, and knowledge distillation, and employ the kernels along deep neural networks (DNNs) to attain rich models for inference. Empirically, we observed the superiority of the kernelization for the representation learning in hyperbolic spaces.

2. Related Work

Geometric Constraint Learning. Geometric constraints have been studied extensively in deep learning, which pushes the network to encode complex structures of the data. The representation power of a set is improved by fitting a subspace [47]. In SVDNet, the orthogonality constraint enforces the fully connected layer lying on the Grassmannian manifold, which de-correlates the features among entries [50]. The works in [39, 41] also show that embedding in a spherical space is particularly effective for similarity learning (*e.g*., face verification, clustering) compared to using Euclidean spaces.

In recent years, hyperbolic geometry has gained substantial interest thanks to its tree-like nature, and the ability to encode hierarchical relationships in the data. Generalizing the basic operations in Euclidean geometry, the work [18] develops hyperbolic layers in neural networks. The following works further show the success of hyperbolic embeddings for graph-structured data, language data, visual data as well as 3D data [38, 21, 32, 4]. More complex structures of data are also studied in [20, 48], which represents the data in a mixed-curvature geometry.

Kernel Methods. Kernel methods have been studied extensively and proven its success in a broad range of machine learning approaches, *e.g*., SVM, PCA and clustering [26]. The main idea of kernel methods is to project the input samples, to a high-dimensional (or even infinite-dimensional) Reproducing Kernel Hilbert Space (RKHS), where the projected data can be analyzed with linear models. To avoid explicit lifting to RKHS, the kernel trick provides a simple way to generate the similarity measure of pairs in RKHS.

As of late, attempts to boost the representational power of structured-data by generalizing the kernel methods to non-linear geometries have gained increasing attention. The common strategy to define a valid pd kernel on non-Euclidean geometries is to adopt a proper distance metric. In [29], the authors propose the main theoretical framework to design the Gaussian kernel on symmetric positive definite matrices. The proposed theory is further verified to develop the Gaussian kernel on the Grassmann manifold [30]. Kernels for the Grassmann manifold are studied in [23]. The kernels using the Fisher information metric are developed for the persistence diagrams in [35]. The closest study to our work is the work of Cho *et al*. [8], which formulates the support vector machine (SVM) in hyperbolic spaces. To facilitate the nonlinear decision boundaries, the kernel SVM for the hyperbolic space is also introduced in [8]. However, the proposed indefinite kernel is not universal and hence violates the universal approximation property [42].

In contrast to existing works, our work develops the theoretical framework for positive definite kernels on the hyperbolic geometry. As a complementary concept to the indefinite kernel, our work kernelizes the hyperbolic space, and thus to embed hyperbolic data into a high, possibly infinite, dimensional Hilbert space. In the remainder of this paper, we will present the developed theory and evaluate the algorithms across different challenging applications.

3. Preliminaries and Background

3.1. Notations

Formally, we use $\mathbb{H}^n$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$ and $\mathcal{H}$ to denote n-dimensional hyperbolic spaces, n-dimensional Euclidean spaces, the space of $m \times n$ real matrices and Hilbert spaces. Throughout the paper, the matrices and vectors are denoted by bold capital letters (*e.g.*, $\boldsymbol{X}$) and bold lower-case letters (*e.g.*, $\boldsymbol{x}$), respectively. The transpose of a matrix (*e.g.*, $\boldsymbol{X}$) or a vector (*e.g.*, $\boldsymbol{x}$) is denoted by the superscript $\top$, *e.g.* $\boldsymbol{X}^\top$ or $\boldsymbol{x}^\top$. $\tanh(\cdot) : \mathbb{R} \to \mathbb{R}, \tanh(x) \coloneqq \frac{e^{2x}-1}{e^{2x}+1}$ refers to the hyperbolic tangent function.

3.2. Hyperbolic Geometry

An n-dimensional hyperbolic space $\mathbb{H}^n$ is a Riemannian manifold with a constant negative curvature [1]. The Poincaré ball is a model of n-dimensional hyperbolic geometry in which all points are embedded within an n-dimensional sphere (or inside a circle in the 2D case which is called the Poincaré disk model). Formally, the Poincaré ball model, with curvature c, is defined as a manifold $\mathbb{D}_c^n = \{\boldsymbol{z} \in \mathbb{R}^n : c\|\boldsymbol{z}\| < 1\}$, with the Riemannian metric $g_c^{\mathbb{D}}(\boldsymbol{z}) = \lambda_c^2(\boldsymbol{z}) \cdot g^E$, in which $\lambda_c(\boldsymbol{z})$ is the conformal factor, defined as $\frac{2}{1-c\|\boldsymbol{z}\|^2}$, and $g^E = \mathbf{I}_n$ is the Euclidean metric tensor. Furthermore and to facilitate vector operations, the Möbius gyrovector space may come in handy. The *Möbius addition* for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathbb{D}_c^n$ is defined as:

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

The *geodesic distance* on $\mathbb{D}_c^n$ is:

$$d_c(\boldsymbol{z}_i, \boldsymbol{z}_j) = \frac{2}{\sqrt{c}}\tanh^{-1}(\sqrt{c}\| - \boldsymbol{z}_i \oplus_c \boldsymbol{z}_j\|). \quad (2)$$

For a point $\boldsymbol{z} \in \mathbb{D}_c^n$, the tangent space at $\boldsymbol{z}$, denoted by $T_{\boldsymbol{z}}\mathbb{D}_c^n$, is an inner product space, which contains the tangent vector with all possible directions at $\boldsymbol{z}$. The Riemannian metric $g_c^{\mathbb{D}}$ at point $\boldsymbol{z}$ is a positive definite symmetric bilinear function on $T_{\boldsymbol{z}}\mathbb{D}_c^n$ as $g_c^{\mathbb{D}}(\boldsymbol{z}) : (T_{\boldsymbol{z}}\mathbb{D}_c^n \times T_{\boldsymbol{z}}\mathbb{D}_c^n) \to \mathbb{R}$. The *exponential map* provides a way to project a point $\boldsymbol{p} \in T_{\boldsymbol{z}}\mathbb{D}_c^n$ to the Poincaré ball $\mathbb{D}_c^n$, as follows:

$$\Gamma_{\boldsymbol{z}}(\boldsymbol{p}) = \boldsymbol{z} \oplus_c \big(\tanh(\sqrt{c}\frac{\lambda_c(\boldsymbol{z}) \cdot \|\boldsymbol{p}\|}{2})\frac{\boldsymbol{p}}{\sqrt{c}\|\boldsymbol{p}\|}\big). \quad (3)$$

The inverse process is termed *logarithm map*, which projects a point $\boldsymbol{q} \in \mathbb{D}_c^n$, to the tangent plane of $\boldsymbol{z}$, given as:

$$\Upsilon_{\boldsymbol{z}}(\boldsymbol{q}) = \frac{2}{\sqrt{c}\lambda_c(\boldsymbol{z})}\tanh^{-1}(\sqrt{c}\| - \boldsymbol{z} \oplus_c \boldsymbol{q}\|)\frac{-\boldsymbol{z} \oplus_c \boldsymbol{q}}{\| - \boldsymbol{z} \oplus_c \boldsymbol{q}\|}. \quad (4)$$

Note that $\Upsilon_{\boldsymbol{z}}\big(\Gamma_{\boldsymbol{z}}(\boldsymbol{p})\big) = \boldsymbol{p} \in T_{\boldsymbol{z}}\mathbb{D}_c^n$. Both the exponential and the logarithm maps are injective functions. In this paper, we leverage the Euclidean space in the identity tangent plane to define the kernels for hyperbolic spaces.

4. Kernel Methods in Hyperbolic Spaces

In this section, we propose positive definite (pd) kernels in hyperbolic spaces. Essentially, we are interested in identifying a bivariate function $k(\cdot,\cdot) : (\mathbb{D}_c^n \times \mathbb{D}_c^n) \to \mathbb{R}$, which represents an inner product in a Reproducing Kernel Hilbert Space (RKHS). Obviously, not all bivariate functions constitute valid kernels, meaning that they do not necessarily realize an RKHS. Also, popular kernels in Euclidean spaces cannot lead to meaningful solutions as they are not faithful to the geometry of the hyperbolic spaces. Embedding hyperbolic points into an RKHS is not only theoretically appealing but can also result in practical benefits due to the intriguing properties of RKHSs. This includes representational power of RKHS [26], kernel two-sample test [19], neural tangent kernels [28] to name a few.

In this paper, we make use of the tangent space of the hyperbolic geometry to define a set of valid pd kernels. We start by formally defining a pd kernel.

Definition 1. *(Positive Definite Kernels [3]) Let $\mathcal{Z}$ be a non-empty set. A symmetric function $k(\cdot,\cdot) : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ is a positive definite kernel on $\mathcal{Z}$ if and only if $\sum_{i,j=1}^m c_i c_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) \geq 0$ for any $m \in \mathbb{N}$, $\boldsymbol{z}_i \in \mathcal{Z}$ and $c_i \in \mathbb{R}$.*

Essential to our work is the following lemma;

Lemma 1. *Let $\mathcal{Z}$ be a non-empty set. Consider a function $f(\cdot) : \mathcal{Z} \to \mathbb{R}^n$, that maps each element of $\mathcal{Z}$ uniquely to $\mathbb{R}^n$. Then,*

$$k(\boldsymbol{z}_i, \boldsymbol{z}_j) = \big\langle f(\boldsymbol{z}_i), f(\boldsymbol{z}_j)\big\rangle$$

is a pd kernel on $\mathcal{Z}$.

Proof. The proof of this lemma follows immediately from Definition 1. To see this, define

$$\boldsymbol{F}_{n\times m} \coloneqq \big[f(\boldsymbol{z}_1), f(\boldsymbol{z}_2), \cdots, f(\boldsymbol{z}_m)\big] .$$

Now, notice that

$$\sum_{i,j=1}^m c_i c_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) = \boldsymbol{c}^\top \boldsymbol{K}\boldsymbol{c} = \boldsymbol{c}^\top \boldsymbol{F}^\top \boldsymbol{F}\boldsymbol{c} = \|\boldsymbol{F}\boldsymbol{c}\|^2 \geq 0 .$$

The $\big[\boldsymbol{K}_{m\times m}\big]_{i,j} = k(\boldsymbol{z}_i, \boldsymbol{z}_j)$ is called the gram matrix. □

Based on Lemma 1, we propose to make use of $f_{\mathbb{D}}(\cdot) : \mathbb{D}_c^n \to \mathbb{R}^n$ defined as,

$$f_{\mathbb{D}}(\boldsymbol{z}) \coloneqq \tanh^{-1}(\sqrt{c}\|\boldsymbol{z}\|)\frac{\boldsymbol{z}}{\sqrt{c}\|\boldsymbol{z}\|}, \quad (5)$$

to develop valid pd kernels on $\mathbb{D}_c^n$. The function $f_{\mathbb{D}}(\cdot)$ enjoys various unique properties. First note that the function is bijective and $f_{\mathbb{D}}(\boldsymbol{z}) = \Upsilon_{\mathbf{0}}(\boldsymbol{z})$. The next theorem establishes an important property and justifies our choice here better.

Theorem 1 (Curve Length Equivalence)**.** *A curve in $\mathbb{D}_c^n$ is a continuous function $\gamma(\cdot) : [0,1] \to \mathbb{D}_c^n$; joining the starting point $\gamma(0)$ to the end point $\gamma(1)$. Define the distance induced by $f_{\mathbb{D}}$ as*

$$d_e(\boldsymbol{z}_i, \boldsymbol{z}_j) := \|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\|. \tag{6}$$

The length of any given curve γ is the same under d_e and the geodesic distance d_c up to a scale of $1/\tilde{\lambda}_c$, where $\tilde{\lambda}_c = 2$ is the conformal factor at the origin.

Proof. The proof is relegated to the supplementary material of our paper due to space limitations. □

Having $f_{\mathbb{D}}$ at our disposal, we are now ready to define the kernels in hyperbolic spaces.

4.1. Hyperbolic Tangent Kernel

The simplest pd kernel resembles the linear kernel in Euclidean spaces and is defined as $k^{\tan}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \langle f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j)\rangle$. We call this kernel hyperbolic tangent kernel as it can be understood as the linear kernel in the identity tangent space of the Poincaré ball. This kernel is attractive as it is parameter-less, making it ideal for fast prototyping. The proof of positive-definiteness of the hyperbolic tangent kernel follows directly from Lemma 1.

4.2. Hyperbolic RBF Kernel

The Gaussian RBF kernel is a popular universal kernel in Euclidean spaces. In $\mathbb{R}^n$, the RBF kernel can be written as $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\xi\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$, $\xi > 0$, where the metric is the squared Euclidean distance in $\mathbb{R}^n$. Taking into account the properties of the RBF kernel [9], it is very desirable to extend this kernel to hyperbolic spaces. One may assume that replacing the Euclidean distance by the geodesic distance (*i.e.*, Eq. (2)) can lead to a valid pd kernel. This, unfortunately, is not the case as shown by the toy example below.

Example 1. *Consider $\mathbb{D}_{0.1}^3$ and the following points:*

$$\boldsymbol{z}_1 = \begin{bmatrix}0.1885\\0.2330\\0.9526\end{bmatrix}, \boldsymbol{z}_2 = \begin{bmatrix}0.6586\\0.2053\\0.0894\end{bmatrix}, \boldsymbol{z}_3 = \begin{bmatrix}0.3017\\0.4155\\0.5357\end{bmatrix}, \boldsymbol{z}_4 = \begin{bmatrix}0.2388\\0.8290\\0.3790\end{bmatrix}.$$

The gram matrix (i.e., $\exp(-\xi d_c^2(\boldsymbol{z}_i, \boldsymbol{z}_j))$ *for* $\xi = 0.01$) *for these points has a negative eigenvalue of* -3.0605×10^{-5}.

Further to the counterexample above, the RBF kernel derived from the geodesic distance is shown to be pd iff the space is isometric to the Euclidean space per the following theorem.

Theorem 2 (Theorem 6.2 in [30])**.** *Let $\mathcal{M}$ be a complete Riemannian manifold and $d_{\mathcal{M}}$ be the induced geodesic distance on the manifold. The Gaussian RBF kernel $k(\cdot,\cdot) : (\mathcal{M} \times \mathcal{M}) \to \mathbb{R} : k(\boldsymbol{m}_i, \boldsymbol{m}_j) := \exp(-\xi d_{\mathcal{M}}^2(\boldsymbol{m}_i, \boldsymbol{m}_j))$ is positive definite for all $\xi > 0$ if and only if the Riemannian manifold $\mathcal{M}$ is isometric to some Euclidean space $\mathbb{R}^n$.*

According to Theorem 2, it is theoretically impossible to obtain a valid RBF kernel using geodesic distance on hyperbolic spaces [1]. Given the above, we propose to make use of $d_e(\cdot,\cdot)$ and define the hyperbolic RBF kernel as

$$k^{\mathrm{rbf}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\xi\|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\|^2\big). \tag{7}$$

To show that the form in Eq. (7) is a valid pd kernel, we first define negative definite (nd) kernels.

Definition 2 (Negative Definite Kernels [3])**.** *Let $\mathcal{Z}$ be a non-empty set. A symmetric function $k(\cdot,\cdot) : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ is a negative definite kernel on $\mathcal{Z}$ if and only if $\sum_{i,j=1}^m c_i c_j k(\boldsymbol{z}_i, \boldsymbol{z}_j) \leq 0$ for any $m \in \mathbb{N}$, $\boldsymbol{z}_i \in \mathcal{Z}$ and $c_i \in \mathbb{R}$ with $\sum_{i=0}^m c_i = 0$.*

Note the difference between pd and nd kernels. For nd kernels, an additional condition (*i.e.*, $\sum_{i=0}^m c_i = 0$) is required. The following lemma shows that $d_e^2(\cdot,\cdot) = \|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\|^2$ is indeed nd.

Lemma 2. *Let $\mathcal{Z}$ be a non-empty set. An injective function $f(\cdot) : \mathcal{Z} \to \mathbb{R}^n$, maps each vector in $\mathcal{Z}$ onto an inner product space $\mathbb{R}^n$. Then $k(\boldsymbol{z}_i, \boldsymbol{z}_j) := \|f(\boldsymbol{z}_i) - f(\boldsymbol{z}_j)\|^2$ is negative definite.*

Proof. The proof is relegated to the supplementary material of our paper due to space limitations. □

The following important theorem establishes the connection between positive definite kernels and negative definite kernels.

Theorem 3. *([3]) Let $\mathcal{Z}$ be a non-empty set and $k : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ be a kernel. The kernel $k(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp(-\xi\Phi(\boldsymbol{z}_i, \boldsymbol{z}_j))$ is positive definite for all $\xi > 0$ if and only if $\Phi(\cdot,\cdot)$ is negative definite.*

Stating the fact that $d_e^2(\cdot,\cdot)$ is nd along with Theorem 3 concludes our claim that the hyperbolic RBF kernel defined in Eq. (7) is pd.

4.3. Hyperbolic Laplace Kernel

The Laplace kernel is another widely used universal kernel in Euclidean spaces, formulated as $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\xi\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$, $\xi > 0$. When extending the Laplace kernel to hyperbolic spaces, we use the following theorem to build a nd kernel for hyperbolic spaces.

[1] If a manifold $\mathcal{M}$ is isometric to some Euclidean spaces $\mathbb{R}^n$, then the geodesic distance on $\mathcal{M}$ is the Euclidean distance in $\mathbb{R}^n$. However, it is impossible to find an isometry between $\mathbb{D}_c^n$ and $\mathbb{R}^n$ because of the difference in the curvature of two geometries.

Theorem 4. *([3]) If $k : (\mathcal{Z} \times \mathcal{Z}) \to \mathbb{R}$ is negative definite and satisfies $k(\boldsymbol{z}_i, \boldsymbol{z}_j) \geq 0$, then k^α is also negative definite for $0 < \alpha < 1$.*

Combining Theorem 3 and Theorem 4, and choosing $\alpha = \frac{1}{2}$, we could obtain the hyperbolic Laplace kernel as $k^{\text{lap}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\xi d_e(f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j))\big) = \exp\big(-\xi \|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\|\big)$. A more general form of the Laplace kernel (*i.e.*, generalized hyperbolic Laplace kernel) can be further derived as: $k^{\text{glap}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\big(-\xi \|f_{\mathbb{D}}(\boldsymbol{z}_i) - f_{\mathbb{D}}(\boldsymbol{z}_j)\|^{2\alpha}\big)$, where $0 < \alpha < 1$.

4.4. Hyperbolic Binomial Kernel

In addition to the exponential type kernels, we further construct a hyperbolic binomial kernel. To obtain the hyperbolic binomial kernel, we make use of the following lemma.

Lemma 3. *Let $\mathcal{Z}$ be a non-empty set. An injective function $f : \mathcal{Z} \to \mathbb{R}^n$, maps each vector in $\mathcal{Z}$ onto an inner product space $\mathbb{R}^n$. Then $k(\boldsymbol{z}_i, \boldsymbol{z}_j) := \big(1 - \langle f(\boldsymbol{z}_i), f(\boldsymbol{z}_j)\rangle\big)^{-\alpha}$ defines a binomial kernel on $\mathcal{Z}$ when $\alpha > 0$ and $\|f(\boldsymbol{z})\| < 1$.*

Proof. According to Lemma 4.8 of [9], if the function $k(\cdot, \cdot)$ can be decomposed by a full Taylor series with each term being non-negative, then we can claim $k(\cdot, \cdot)$ is a valid pd kernel. Let $t = \langle f(\boldsymbol{z}_i), f(\boldsymbol{z}_j)\rangle$, the binomial series $k(\boldsymbol{z}_i, \boldsymbol{z}_j) = (1 - t)^{-\alpha} = \sum_{n=0}^{\infty} \binom{-\alpha}{n}(-1)^n t^n$ holds for all $|t| < 1$, where the binomial coefficient $\binom{\beta}{n} := \prod_{i=1}^{n}(\beta - i + 1)/i$. It can be seen $\binom{-\alpha}{n}(-1)^n > 0$ when $\alpha > 0$, which indicates the binomial kernel has a non-negative and full Taylor series. □

According to the Lemma 3, we could obtain the hyperbolic binomial kernel as

$$k^{\text{bin}}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \big(1 - \langle f_{\mathbb{D}}(\boldsymbol{z}_i), f_{\mathbb{D}}(\boldsymbol{z}_j)\rangle\big)^{-\alpha},\ \alpha > 0. \quad (8)$$

Also, given the non-negativeness and full Taylor series in the above proof, we can further claim that the hyperbolic binomial kernel satisfies the necessary and sufficient condition of being universal, shown in Corollary 4.57 of [9].

Remark 1. *As alluded to earlier, we have made use of the identity tangent space of the Poincaré ball* (i.e., $\mathbb{D}_c^n$) *to define pd kernels for the hyperbolic spaces. This implies that the kernels are defined using the Lie algebra of $\mathbb{D}_c^n$. Such a construction has been used with success in other manifolds* (e.g., *SPD as in [30]).*

In this paper, we employ the kernels along with convolutional neural networks (CNNs) to attain rich models for computer vision tasks. The CNNs encode the input data to vectors, distributed in hyperbolic spaces. Then the proposed kernels are further used to train the network.

5. Experiments

We first explain the inference with cross entropy-like loss function using kernels. Specifically, for a training sample $\boldsymbol{f}_i$ with label l, the cross entropy loss is given by:

$$\mathcal{L} = -\log\Big(\frac{\exp(s(\boldsymbol{f}_i, \boldsymbol{w}_l))}{\sum_{j=1}^{N}\exp(s(\boldsymbol{f}_i, \boldsymbol{w}_j))}\Big), \quad (9)$$

where $\boldsymbol{w}_i$ indicates the weights or prototype for $\boldsymbol{f}_i$ and N is the number of classes in the dataset. Then we apply our kernels in Eq. (9) as:

$$\mathcal{L}^K = -\log\Big(\frac{g(k(\boldsymbol{f}_i, \boldsymbol{w}_l))}{\sum_{j=1}^{N} g(k(\boldsymbol{f}_i, \boldsymbol{w}_j))}\Big). \quad (10)$$

Here, $g(\cdot)$ is $\exp$ mapping if $k(\cdot, \cdot)$ is non-exponential type kernels. Otherwise, $g(\cdot)$ is the identity mapping.

In the remainder of this section, we comprehensively evaluate the effectiveness of the proposed algorithms for a variety of challenging tasks, *i.e.*, few-shot learning, zero-shot learning, person re-identification and knowledge distillation. Full details of all experiments done in this paper are provided in the supplementary material.

5.1. Few-shot Learning

Few-shot learning (FSL) is required to learn an embedding space, which should be adapted to recognize unseen classes at test time, given only a few samples of each new class [49, 27]. In our experiments, we follow the general practice (*i.e.*, 5-way 1-shot and 5-way 5-shot and 15 query images) to evaluate the model. We employ the pipeline in the prototypical network (ProtoNet) [49] along with the proposed kernels to train the feature extractor.

In terms of the feature extractor, we use both Conv-4 [49] and ResNet-18 [24] CNN backbones in our experiments. Moreover, four popular benchmarks, *i.e.*, ***mini*ImageNet** [11], **CUB** [53], ***tiered*-ImageNet** [45] and **Few-shot-CIFAR100** (FC100) [43] are adopted to assess our algorithms. We use the Conv-4 and ResNet-18 backbones to evaluate the *mini*ImageNet and CUB datasets and the Conv-4 backbone to evaluate the *tiered*-ImageNet and FC100 datasets. Please refer to the supplementary material for more details about the statistics of each dataset and the implementation details.

Tables 2, 3, 4 illustrate the results on four datasets. We observe that our algorithms improve the few-shot recognition performance as compared to their hyperbolic counterpart and other advanced methods. In addition, the results from the hyperbolic RBF kernel in general exceed the results from other kernels. For example, in 5-way 5-shot setting, the hyperbolic RBF kernel outperforms the Hyperbolic ProtoNet [32] by 3.42, 2.68, 4.52 and 2.64 for *mini*ImageNet, CUB, *tiered*-ImageNet and FC100, respectively, clearly showing the potential and superiority of universal kernels.

Table 2. Few-shot classification results on the *mini*ImageNet dataset with 95% confidence interval.

Model	Backbone	1-shot	5-shot
MatchingNet [52]	Conv-4	43.56 ± 0.84	55.31 ± 0.73
ProtoNet [49]	Conv-4	44.53 ± 0.76	65.77 ± 0.66
MAML [16]	Conv-4	48.70 ± 1.84	63.11 ± 0.92
RelationNet [51]	Conv-4	50.44 ± 0.82	65.32 ± 0.70
DN4 [37]	Conv-4	51.24 ± 0.74	71.02 ± 0.64
DSN [47]	Conv-4	51.78 ± 0.96	68.99 ± 0.69
Hyper ProtoNet [32]	Conv-4	54.43 ± 0.20	72.67 ± 0.15
Hyperbolic tangent kernel	Conv-4	55.61 ± 0.21	74.81 ± 0.16
Hyperbolic RBF kernel	Conv-4	56.48 ± 0.20	**76.09 ± 0.16**
Hyperbolic Laplace kernel	Conv-4	56.26 ± 0.20	75.35 ± 0.15
Hyperbolic binomial kernel	Conv-4	**56.82 ± 0.20**	75.27 ± 0.15
Baseline [6]	ResNet-18	51.75 ± 0.80	74.27 ± 0.63
Baseline++ [6]	ResNet-18	51.87 ± 0.77	75.68 ± 0.63
MatchingNet [52]	ResNet-18	52.91 ± 0.88	68.88 ± 0.69
ProtoNet [49]	ResNet-18	54.16 ± 0.82	73.68 ± 0.65
SNCA [54]	ResNet-18	57.80 ± 0.80	72.80 ± 0.70
Hyper ProtoNet [32]	ResNet-18	59.47 ± 0.20	76.84 ± 0.14
Hyperbolic tangent kernel	ResNet-18	59.91 ± 0.21	76.65 ± 0.16
Hyperbolic RBF kernel	ResNet-18	60.91 ± 0.21	77.12 ± 0.15
Hyperbolic Laplace kernel	ResNet-18	60.52 ± 0.21	**77.33 ± 0.15**
Hyperbolic binomial kernel	ResNet-18	**61.04 ± 0.21**	77.01 ± 0.15

Table 3. Few-shot classification results on the CUB dataset with 95% confidence interval. † indicates the network was self-implemented.

Model	Backbone	1-shot	5-shot
MatchingNet [52]	Conv-4	61.16 ± 0.89	72.86 ± 0.70
ProtoNet [49]	Conv-4	51.31 ± 0.91	70.77 ± 0.69
MAML [16]	Conv-4	55.92 ± 0.95	72.09 ± 0.76
RelationNet [51]	Conv-4	62.45 ± 0.98	76.11 ± 0.69
DN4 [37]	Conv-4	53.15 ± 0.84	81.90 ± 0.60
Hyper ProtoNet [32]	Conv-4	64.02 ± 0.20	82.53 ± 0.14
Hyperbolic tangent kernel	Conv-4	66.14 ± 0.23	82.11 ± 0.15
Hyperbolic RBF kernel	Conv-4	**70.98 ± 0.22**	**85.21 ± 0.13**
Hyperbolic Laplace kernel	Conv-4	68.27 ± 0.23	84.64 ± 0.13
Hyperbolic binomial kernel	Conv-4	69.05 ± 0.23	83.00 ± 0.14
Baseline [6]	ResNet-18	65.51 ± 0.87	82.85 ± 0.55
Baseline++ [6]	ResNet-18	67.02 ± 0.77	83.58 ± 0.54
RelationNet [51]	ResNet-18	67.59 ± 0.58	82.75 ± 0.58
MAML [16]	ResNet-18	69.96 ± 1.01	82.70 ± 0.65
ProtoNet [49]	ResNet-18	71.88 ± 0.91	86.64 ± 0.51
MatchingNet [52]	ResNet-18	72.36 ± 0.90	83.64 ± 0.60
Hyper ProtoNet† [32]	ResNet-18	72.86 ± 0.22	85.69 ± 0.13
Hyperbolic tangent kernel	ResNet-18	73.52 ± 0.22	88.75 ± 0.11
Hyperbolic RBF kernel	ResNet-18	**75.79 ± 0.21**	**89.98 ± 0.11**
Hyperbolic Laplace kernel	ResNet-18	74.37 ± 0.21	89.08 ± 0.12
Hyperbolic binomial kernel	ResNet-18	74.46 ± 0.22	89.28 ± 0.11

Table 4. Few-shot classification results on *tiered*-ImageNet and FC100 datasets with 95% confidence interval. † indicates the network was self-implemented.

Model	*tiered*-ImageNet		FC100	
	1-shot	5-shot	1-shot	5-shot
Hyper ProtoNet† [32]	54.44 ± 0.23	71.96 ± 0.20	37.59 ± 0.19	51.76 ± 0.19
Hyperbolic tangent kernel	54.73 ± 0.22	74.37 ± 0.18	37.66 ± 0.17	52.29 ± 0.18
Hyperbolic RBF kernel	**57.78 ± 0.23**	76.11 ± 0.18	**38.93 ± 0.18**	**54.40 ± 0.18**
Hyperbolic Laplace kernel	57.33 ± 0.22	**76.48 ± 0.18**	37.99 ± 0.17	53.54 ± 0.18
Hyperbolic binomial kernel	56.72 ± 0.22	75.87 ± 0.18	38.32 ± 0.18	53.50 ± 0.18

5.2. Zero-shot Learning

Zero-shot learning (ZSL) aims to identify objects that are unseen during the training phase [2, 55]. We first build a baseline network for the scenario of zero-shot recognition. In the training phase, we randomly sample N_b seen visual features as $\boldsymbol{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_{N_b}\}$. All the semantic features are projected to the visual space, denoted by $\boldsymbol{E} = \{e(\boldsymbol{a}_1), \dots, e(\boldsymbol{a}_{|L^s|})\}$, where $|L^s|$ denotes the number of seen classes in the training set. In our implementation, the embedding function (*i.e.*, $e(\cdot)$) is a simple two layer MLP, with each layer stacking the linear transformation, ReLU activation and batch normalization. Then the network is trained by the following cross-entropy type loss:

$$\mathcal{L}_{\text{zsl}} = -\frac{1}{N_b}\sum_{i=1}^{N_b} \log\Big(\frac{\exp\big(-\|(e(\boldsymbol{a}^*) - \boldsymbol{v}_i\|\big)}{\sum_{j=1}^{|L^s|}\exp\big(-\|e(\boldsymbol{a}_j) - \boldsymbol{v}_i\|\big)}\Big),$$

where $\boldsymbol{a}^*$ shares the same label with $\boldsymbol{v}_i$. The baseline network is conducted on Euclidean spaces. The usage of the kernels for ZSL is detailed in the supplementary material.

Four datasets, *i.e.*, **SUN** [44], **CUB** [53], **AWA1** [34] and **AWA2** [2] are adopted to evaluate our algorithms in the generalized ZSL (GZSL) setting. We report the top-1 mean class accuracy (MCA) for both the unseen classes (U) and the seen classes (S) and also calculate the harmonic mean (HM) score, *i.e.* $\text{HM} = 2 \times \text{U} \times \text{S}/(\text{U} + \text{S})$. Please refer to the supplementary material for more details about the statistics of each dataset and implementation details.

We first evaluate the effectiveness of our methods by comparing them against the baseline. As shown in Table 5, each hyperbolic kernel brings a significant improvement to the baseline network. For example, the simplest hyperbolic tangent kernel improves the HM value over the baseline by 6.1, 21.6, 21.9 and 14.1 for SUN, CUB, AWA1 and AWA2, respectively. In addition, the powerful hyperbolic RBF kernel or hyperbolic Laplace kernel continues to improve the representation capacity, again showing the superiority of the kernel design for embedding learning.

To further verify the effectiveness of our approach, we continue to compare our methods to a couple of popular ZSL algorithms, including the state-of-the-art non-generative methods [56, 36]. We observe that our hyperbolic RBF kernel and hyperbolic Laplace kernel achieve competitive results to the state-of-the-art methods across four datasets. ZSL is a very challenging task, and while none of the methods in Table 5 achieved the best performance across all four datasets, it is very competitive. Thus, to establish this objectively, we employ the Friedman test[2] [10] to compare the algorithms. As shown in the last column of Table 5, the ranking list clearly shows that our meth-

[2] The Friedman test is a non-parametric measure for multiple datasets. It ranks the algorithms for each dataset separately and calculates the average ranks for each dataset as a ranking score.

Table 5. Zero-shot recognition results on SUN, CUB, AWA1 and AWA2 datasets. U and S indicate the accuracy for unseen and seen classes, respectively. HM is the harmonic mean of U and S.

Model	SUN			CUB			AWA1			AWA2			Friedman
	U	S	HM	U	S	HM	U	S	HM	U	S	HM	test (rank)
LATEM [55]	14.7	28.8	19.5	15.2	57.3	24.0	7.3	71.7	13.3	11.5	77.3	20.0	12.0 (12)
DEVISE [17]	16.9	27.4	20.9	23.8	53.0	32.8	13.4	68.7	22.4	17.1	74.7	27.8	10.0 (11)
DEM [57]	20.5	34.3	25.6	19.6	57.9	29.2	32.8	84.7	47.3	30.5	86.4	45.1	9.33 (9)
ALE [2]	21.8	33.1	26.3	23.7	62.8	34.4	16.8	76.1	27.5	14.0	81.8	23.9	9.33 (9)
SP-AEN [5]	24.9	38.6	30.3	34.7	70.6	46.6	-	-	-	23.3	**90.9**	37.1	7.67 (7)
CRnet [56]	34.1	36.5	35.3	45.5	56.8	50.5	58.1	74.7	65.4	52.6	78.8	63.1	3.00 (4)
Kai *et al.* [36]	36.3	42.8	39.3	**47.4**	47.6	47.5	**62.7**	77.0	69.1	**56.4**	81.4	**66.7**	2.83 (3)
Baseline	22.8	38.0	28.5	18.6	44.6	26.3	29.8	76.4	42.9	25.5	76.4	38.2	8.67 (8)
Hyperbolic tangent kernel	29.4	42.0	34.6	40.8	**58.1**	47.9	52.3	85.2	64.8	37.1	88.5	52.3	5.00 (5)
Hyperbolic RBF kernel	**37.0**	**43.3**	**39.9**	44.6	57.8	50.3	59.0	84.6	69.5	42.9	89.5	57.9	2.67 (2)
Hyperbolic Laplace kernel	35.1	44.2	39.1	46.2	56.1	**50.7**	60.7	83.5	**70.3**	54.1	87.1	**66.7**	1.83 (1)
Hyperbolic binomial kernel	26.9	43.8	33.3	39.8	56.9	46.8	43.7	**88.9**	58.6	39.8	90.5	55.4	5.67 (6)

ods with the hyperbolic Laplace kernel and the hyperbolic RBF kernel are the best two options in general for the ZSL task.

5.3. Person Re-identification

Person re-identification (re-ID) is an important application in the video/multi-camera surveillance task [13, 12]. Following the work [32], ResNet-50, pre-trained on ImageNet, is employed as a backbone network and we also perform experiments across three dimensions, *i.e.*, 32, 64, 128, for the feature representation. Both **Market-1501** [58] and **DukeMTMC-reID** [46] pedestrian datasets are used to evaluate our approaches. We use both mean average precision (mAP) and rank-1 accuracy of cumulative matching characteristic (CMC) to evaluate our algorithms. Different from FSL and ZSL, we use the generalized hyperbolic Laplace kernel in the re-ID experiment, as we observe that the generalized hyperbolic Laplace kernel achieves fairly good performance compared to the hyperbolic Laplace one. Please refer to the supplementary material for more details.

We compare the proposed algorithms to the methods in [32]. As shown in Table 6, we observe that our algorithms bring positive effects to the retrieval performance on both datasets, especially for the mAP value. In the market-1501 dataset, most of our methods achieve competitive performance compared to [32]. However, we also observe that the binomial kernel cannot perform well in different embedding sizes. In the DukeMTMC-reID dataset, our method could outperform its hyperbolic counterpart on both R-1 and mAP values and the RBF kernel is the most powerful one, which is superior to the other kernels in every dimension. For example, the hyperbolic RBF kernel improves the R-1 / mAP values over the work [32] by 5.1 / 6.6, 3.0 / 7.2 and 1.9 / 6.8 for the dimension of 32, 64 and 128, respectively.

Table 6. Person re-ID results on Market-1501 and DukeMTMC-reID datasets. The value in $\boxed{\cdot}$ denotes the result below the performance in [32]. g-Hyperbolic Laplace kernel indicates the generalized hyperbolic Laplace kernel.

Model	Dim	Market-1501		DukeMTMC-reID	
		R-1	mAP	R-1	mAP
Euclidean [32]	#32	68.0	43.4	57.2	35.7
Hyperbolic [32]	#32	75.9	51.9	62.2	39.1
Hyperbolic tangent kernel	#32	$\boxed{75.4}$	53.3	63.9	42.5
Hyperbolic RBF kernel	#32	76.0	54.3	**67.3**	**46.3**
g-Hyperbolic Laplace kernel	#32	**78.7**	**56.3**	64.1	40.7
Hyperbolic binomial kernel	#32	$\boxed{75.2}$	55.0	63.7	44.7
Euclidean [32]	#64	80.5	57.8	68.3	45.5
Hyperbolic [32]	#64	84.4	62.7	70.8	48.6
Hyperbolic tangent kernel	#64	**85.8**	68.0	**73.9**	54.2
Hyperbolic RBF kernel	#64	85.2	65.7	73.8	**55.8**
g-Hyperbolic Laplace kernel	#64	85.4	**68.4**	73.3	50.6
Hyperbolic binomial kernel	#64	$\boxed{83.0}$	64.6	71.5	54.0
Euclidean [32]	#128	86.0	67.3	74.1	53.3
Hyperbolic [32]	#128	87.8	68.4	76.5	55.4
Hyperbolic tangent kernel	#128	**89.4**	**74.1**	**78.6**	60.9
Hyperbolic RBF kernel	#128	88.9	73.5	78.4	**62.2**
g-Hyperbolic Laplace kernel	#128	$\boxed{87.6}$	72.4	77.3	59.6
Hyperbolic binomial kernel	#128	$\boxed{87.6}$	72.0	$\boxed{75.4}$	59.2

5.4. Knowledge Distillation

Knowledge distillation (KD) is an efficient method to train a small student network, under the supervision of a pre-trained larger teacher network [25, 7]. In the teacher-student network, the output of the teacher network acts as ground truth to train a student network. For a training image (*e.g.*, $\boldsymbol{X}$), the teacher network and student network generate the prediction scores $\boldsymbol{g} = [g_1, g_2, \ldots, g_N]$ and $\boldsymbol{p} = [p_1, p_2, \ldots, p_N]$, respectively. Noted that $\boldsymbol{g}$ and $\boldsymbol{p}$ are normalized by the $\mathrm{softmax}$ function. Then the KD loss is given by: $\mathcal{L}_{kd} = -\sum_{i=1}^{N} g_i \mathrm{log}(p_i)$.

We use the ResNet-20 as a teacher network and a simple 4-layer CNN as a student network. We report the results on **CIFAR-10** and **CIFAR-100** benchmarks [33]. The details of the usage of kernels, network architecture and datasets are summarized in the supplementary material. We use the top-1 mean accuracy to evaluate the networks. Please refer to the supplementary material for more details about the network training and corresponding hyper-parameters. As shown in Table 7, we can again find that our hyperbolic kernels improve the accuracy over the baseline, and the hyperbolic RBF kernel brings the maximum performance gain, 3.1 / 4.5 for CIFAR-10 / CIFAR-100, respectively.

Table 7. Knowledge distillation results on CIFAR-10 / 100 datasets. g-Hyperbolic Laplace kernel indicates the generalized hyperbolic Laplace kernel.

Model	CIFAR-10	CIFAR-100
Baseline	80.5	49.9
Hyperbolic tangent kernel	82.1	50.5
Hyperbolic RBF kernel	**83.6**	**54.4**
g-Hyperbolic Laplace kernel	83.2	53.9
Hyperbolic binomial kernel	81.6	51.8

5.5. Further Studies

To the best of our knowledge, our work is the first to develop pd kernels in hyperbolic spaces. That said, indefinite hyperbolic kernels are developed in [8]. We compare and contrast the two school of thoughts. In doing so, we consider the problem of few-shot learning and follow the setup of [32]. As for the indefinite kernel, we use the Minkowski inner product kernel, presented in [8] (see supplementary material for details). We have evaluated the performance of our pd kernels and the indefinite kernel for the task of 5-way 5-shot learning across the *mini*ImageNet, CUB, *tired*-ImageNet and FC100 datasets. Fig. 1 shows that the performance attained by the indefinite kernel does not match that of pd kernels, clearly showing the potential of pd kernels for hyperbolic representations.

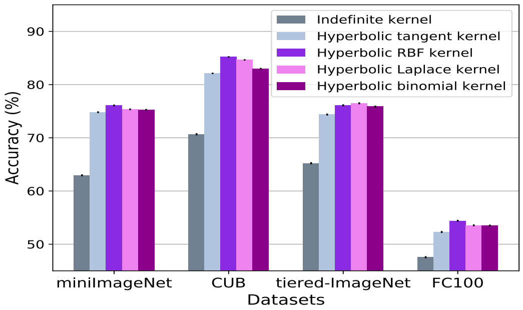

Figure 1. The performance comparison between the indefinite kernel and pd kernels for hyperbolic representations.

One may wonder how useful the hyperbolic spaces are and their kernels in comparison to simple Euclidean kernels. In the end, the Poincaré ball is embedded in n-dimensional Euclidean spaces and hence conventional kernels can be applied seamlessly. In Fig. 2, we compare the proposed kernels against their Euclidean counterparts again on the task of few-shot learning using the *mini*ImageNet dataset. We observe: **(1)** the kernel machines in both Euclidean spaces and hyperbolic spaces bring performance gain to the deep neural network. **(2)** The proposed hyperbolic kernels can outperform the vanilla Euclidean kernels significantly, again showing the reasonable design of the proposed kernels.

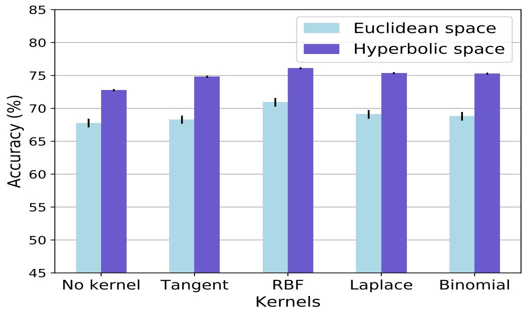

Figure 2. The performance comparison for kernels on Euclidean spaces and Hyperbolic spaces.

Remark 2 (Good Practice of Employing Hyperbolic Geometry)**.** *Few works have studied the problem of learning an embedding in hyperbolic spaces [4, 32]. However, the existing works generate the vectors in the tangent space at the origin and project to the hyperbolic spaces using* $\Gamma_{\mathbf{0}}(\cdot)$ *mapping. A drawback of this framework is that the hyperbolic geometry is not fully utilized as every representation is flattened at the identity. In other words, only the vectors very close to the origin represent hyperbolic distances. In contrast, and in our experiments, we generate hyperbolic representations directly in the Poincaré ball. Empirically, we observe that various applications can benefit from a high curvature* (i.e., *c). For example, in the person re-identification task, the curvature of the Poincaré ball is* 10^{-2} *in our algorithms, while the work in [32] sets it to* 10^{-5}*, which makes the Poincaré ball very flat.*

6. Conclusion

This paper proposes a family of positive definite kernels to embed hyperbolic representations in Hilbert spaces. In such kernels, we leverage the identity tangent space of the Poincaré ball and further define valid positive definite kernels in identity tangent spaces. The proposed kernels include powerful universal kernels (*i.e*., the hyperbolic RBF kernel, the hyperbolic Laplace kernel and the hyperbolic binomial kernel). We evaluate the effectiveness of the kernels in a variety of challenging applications, such as few-shot learning, zero-shot learning, person re-identification and knowledge distillation, and the empirical results have shown positive results for embedding learning via the kernels in hyperbolic spaces. Future works include exploiting the proposed kernels to other applications (*i.e*., natural language processing and graph neural networks). In addition, we have found that the effectiveness of the kernel is data-dependent and we want to develop a rule for choosing the right kernel for a given data.

References

[1] P.-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2007. 3

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *PAMI*, 2015. 6, 7

[3] Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups*. Springer, 1984. 3, 4, 5

[4] Jiaxin Chen, Jie Qin, Yuming Shen, Li Liu, Fan Zhu, and Ling Shao. Learning attentive and hierarchical representations for 3d shape recognition. In *ECCV*, 2020. 2, 8

[5] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, 2018. 7

[6] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 6

[7] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *CVPR*, 2021. 7

[8] Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. Large-margin classification in hyperbolic space. In *ICML*, 2019. 1, 2, 8

[9] Andreas Christmann and Ingo Steinwart. *Support Vector Machines*. Springer, 2008. 4, 5

[10] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 2006. 6

[11] Jia Deng, Wei Dong, Richard Socher Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[12] Pengfei Fang, Jieming Zhou, Soumava Kumar Roy, Pan Ji, Lars Petersson, and Mehrtash Harandi. Attention in attention networks for person retrieval. *PAMI*, 2021. 7

[13] Pengfei Fang, Jieming Zhou, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Bilinear attention networks for person retrieval. In *ICCV*, 2019. 7

[14] Aasa Feragen and Søren Hauberg. Open problem: Kernel methods on manifolds and metric spaces. what is the probability of a positive definite geodesic exponential kernel? In *CoLT*, 2016. 2

[15] Aasa Feragen, François Lauze, and Søren Hauberg. Geodesic exponential kernels: when curvature and linearity conflict. In *CVPR*, 2015. 1

[16] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 6

[17] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. 7

[18] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *NeurIPS*, 2018. 1, 2

[19] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Scholköpf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012. 3

[20] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. Learning mixed-curvature representations in product spaces. In *ICLR*, 2019. 2

[21] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. Hyperbolic attention networks. In *ICLR*, 2019. 2

[22] Matthias Hamann. On the tree-likeness of hyperbolic spaces. *arXiv:1105.3925*, 2011. 1

[23] Mehrtash T. Harandi, Mathieu Salzmann, Sadeep Jayasumana, Richard Hartley, and Hongdong Li. Expanding the family of grassmannian kernels: An embedding perspective. In *ECCV*, 2014. 2

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning Workshop*, 2014. 7

[26] Thomas Hofmann, Bernhard Scholkopf, and Alexander J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 2008. 2, 3

[27] Jie Hong, Pengfei Fang, Weihao Li, Tong Zhang, Christian Simon, Mehrtash Harandi, and Lars Petersson. Reinforced attention for few-shot learning and beyond. In *CVPR*, 2021. 5

[28] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018. 3

[29] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, 2013. 2

[30] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on riemannian manifolds with gaussian RBF kernels. *PAMI*, 2015. 1, 2, 4, 5

[31] Hermann Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 1977. 1

[32] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *CVPR*, 2020. 1, 2, 5, 6, 7, 8

[33] Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Technical report*, 2009. 7

[34] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 2013. 6

[35] Tam Le and Makoto Yamada. Persistence fisher kernel: A riemannian manifold kernel for persistence diagrams. In *NeurIPS*, 2018. 2

[36] Kai Li, Martin Renqiang Min, and Yun Fu. Rethinking zero-shot learning: A conditional visual classification perspective. In *ICCV*, 2019. 6, 7

[37] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Gao Yang, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *CVPR*, 2019. 6

[38] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *NeurIPS*, 2019. 1, 2

[39] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 2

[40] Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the fréchet mean. In *ICML*, 2020. 1

[41] Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. Spherical text embedding. In *NeurIPS*, 2019. 2

[42] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *JMLR*, 2006. 2

[43] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 5

[44] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 6

[45] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *ICLR*, 2018. 5

[46] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016. 7

[47] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *CVPR*, 2020. 2, 6

[48] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *ICLR*, 2020. 2

[49] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 5, 6

[50] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *ICCV*, 2017. 2

[51] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 6

[52] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016. 6

[53] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5, 6

[54] Zhirong Wu, Alexei A Efros, and Stella Yu. Improving generalization via scalable neighborhood component analysis. In *ECCV*, 2018. 6

[55] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 6, 7

[56] Fei Zhang and Guangming Shi. Co-representation network for generalized zero-shot learning. In *ICML*, 2019. 6, 7

[57] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017. 7

[58] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 7