RIEMANNIAN ISOMETRIC KERNELS: PRESERVING INTRINSIC GEOMETRY OF HYPERBOLIC SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Hyperbolic spaces, characterized by constant negative curvature, provide a natural geometric alternative for representing hierarchical data, while kernel methods further enhance the representation power by mapping hyperbolic embeddings into reproducing kernel Hilbert spaces (RKHSs). However, existing hyperbolic kernels enforce the isometry constraint only at the metric-space level, creating a substantial mismatch between the kernel-induced similarity used in practice and the manifold's theoretical geodesic distance. To address this limitation, we introduce Riemannian isometric kernels, whose canonical feature maps induce a metric that coincides exactly with the Riemannian metric of the input manifold, thereby intrinsically preserving its geometric structure. Starting from the definition of Riemannian isometry, we construct a family of Riemannian isometric kernels, including logarithmic kernel (RILog), binomial kernel (RIBin), exponential kernel (RIExp), and integral kernel (RIInt) variants, whose kernel-induced distances used in practice are monotonically increasing functions of the geodesic distance, ensuring consistency between the theoretical formulation and empirical evaluation. We evaluate the proposed kernels through geometric stress analysis and four diverse downstream tasks: zero-shot learning, few-shot learning, graph node classification, and semantic textual similarity. Across all benchmarks, the proposed kernels match or outperform state-of-the-art adaptive hyperbolic kernels, demonstrating their effectiveness and broad applicability.

1 INTRODUCTION

Hyperbolic geometry has emerged as an alternative embedding space for various machine learning applications, including natural language processing (NLP), computer vision (CV), and graph learning (Ganea et al., 2018; Khrulkov et al., 2020; Liu et al., 2019). This is due to its constant negative curvature induces exponential volume growth, making it particularly well suited to representing hierarchical and tree-structured data compared with Euclidean space (Nickel & Kiela, 2017).

To further improve the representational capacity of hyperbolic representations, kernel methods have been extended to hyperbolic spaces. Cho et al. (2019) introduced one of the first kernel SVM formulations for hyperbolic data, while the resulting kernels are not positive definite (PD), which renders the corresponding kernel optimization non-convex and breaks the RKHS-based theoretical guarantees available to PD kernel methods. Therefore, designing valid PD kernels on hyperbolic spaces has attracted sustained interest, as such kernels combine the representation power of kernel machines with the intrinsic geometry of hyperbolic spaces (Fang et al., 2021; 2023; Yang et al., 2023; Si et al., 2026). In Euclidean spaces, valid kernels can be readily constructed from distance functions, as exemplified by the Gaussian RBF kernel. On Riemannian manifolds, however, this construction does not generally extend, because the geodesic exponential kernel is positive definite if and only if the manifold is isometric to a Euclidean space (Feragen et al., 2015; Jayasumana et al., 2015). Early approaches circumvented this obstruction by mapping hyperbolic points onto a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tangent space and applying Euclidean kernels there (Fang et al., 2021; 2023). Although effective in practice, such mappings introduce geometric distortion because tangent-space approximations preserve the manifold geometry only locally. Consequently, the resulting kernels cannot preserve the intrinsic metric globally.

Subsequent studies have sought to recover geometric consistency at the metric-space level, either by defining an auxiliary distance in the RKHS that matches the hyperbolic geodesic distance (Yang et al., 2023) or by introducing adaptive parameters (Si et al., 2026). However, these approaches rely on auxiliary distance measures rather than the kernel-induced distance which is typically used for similarity evaluation in downstream tasks. Consequently, they do not ensure that the canonical feature map itself preserves the geometry of the input manifold and therefore do not fundamentally eliminate geometric distortion introduced by the embedding. More fundamentally, metric-space isometry constrains pairwise distances but does not directly characterize the local differential geometry of the embedding. In contrast, preserving the Riemannian metric tensor directly controls local geometric quantities such as lengths, angles, and volumes. Moreover, no positive-definite kernel can induce a Hilbert-space distance that is globally isometric to the hyperbolic geodesic distance (Schoenberg, 1938). This impossibility rules out exact global distance preservation under the native RKHS metric, motivating a shift from metric-space isometry to Riemannian isometry.

Motivated by this limitation, we shift our focus to Riemannian isometry. A kernel's canonical feature map embeds the input manifold into an RKHS, with its image forming a potentially curved submanifold. The ambient Hilbert-space inner product then induces a Riemannian metric on this embedded submanifold (Amari & Wu, 1999). This perspective leads to a different question: can we construct kernels whose canonical feature maps define Riemannian isometries between the input manifold and its embedded image, thereby preserving its intrinsic Riemannian geometry rather than merely matching pairwise distances?

In this paper, we answer this question affirmatively by introducing a new family of kernels, termed Riemannian isometric kernels, whose canonical feature maps define Riemannian isometries between the input manifolds and their embedded images in the RKHS. By preserving the Riemannian metric under the canonical feature maps, these kernels retain the intrinsic geometry of the input manifolds and eliminate local geometric distortion induced by the embedding. Furthermore, we establish that the kernel-induced distance used in downstream evaluation is a monotonically increasing function of the intrinsic geodesic distance, ensuring consistency between the theoretical geometry and the distance measure used in practice.

Our contributions are summarized as follows:

1. We introduce Riemannian isometric kernels and provides a practical criterion for identifying Riemannian isometric kernels: any smooth PD kernel whose canonical feature map is a topological embedding and that satisfies the Riemannian isometry equation induces an embedded submanifold in the RKHS that is Riemannian-isometric to the input manifold.
2. We construct four concrete Riemannian isometric kernels for hyperbolic spaces, including the Riemannian isometric logarithmic kernel (RILog), binomial kernel (RIBin), exponential kernel (RIExp), and integral kernel (RIInt), and establish their order-preserving property, showing that the kernel-induced distance is a monotonically increasing function of the hyperbolic geodesic distance.
3. We extensively evaluate the proposed kernels geometrically and on four downstream tasks: generalized zero-shot learning, few-shot learning, graph node classification, and semantic textual similarity. Geometric stress analysis shows that the Riemannian isometric kernels achieve lower geometric distortion, while the tasks demonstrate outperformance over existing hyperbolic kernels.

2 Related Work

Hyperbolic representation learning. Following the seminal work on Poincaré embeddings by (Nickel & Kiela, 2017), hyperbolic geometry has been widely explored in word embeddings (Nickel & Kiela, 2018),

neural networks (Ganea et al., 2018), image classification (Khrulkov et al., 2020), graph learning (Liu et al., 2019; Chami et al., 2019), and few-shot learning (Khrulkov et al., 2020). These applications exploit the fact that the exponential volume growth induced by the negative curvature of hyperbolic space naturally accommodates hierarchical and tree-like structures with substantially lower distortion than Euclidean embeddings (Sarkar, 2012; Hamann, 2018).

Hyperbolic kernel methods. Cho et al. (2019) introduced hyperbolic SVMs based on the Minkowski inner product, but the kernel is indefinite, causing non-convex optimization and numerical instability. Fang et al. (2021; 2023) then proposed a family of positive-definite (PD) hyperbolic kernels (tangent, RBF, Laplace, binomial, radial) by mapping data to the tangent space at the identity. These kernels are universal and PD, but the tangent-space approximation preserves geometry only locally, leading to global geomrtic distortion. To reduce this, Yang et al. (2023) developed curvature-aware kernels that aim to preserve hyperbolic geodesic distances in an RKHS. However, their isometry uses an auxiliary distance rather than the canonical RKHS distance, and the kernel forms remain fixed. More recently, Si et al. (2026) introduced adaptive hyperbolic kernels in de Branges–Rovnyak spaces, using a learnable multiplier for task-aware representation.

3 Preliminaries and Background

3.1 Notations

Throughout this paper, we use $\mathbb{R}^n, \mathbb{H}^n_c$ and $\mathcal{H}$ to denote n-dimensional Euclidean space, the n-dimensional hyperbolic space of curvature $-c$ and a Hilbert space, respectively. We use $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and $\|\cdot\|_{\mathcal{H}}$ to denote the inner product and the norm in RKHS $\mathcal{H}$. Vectors are written in bold (e.g., $\boldsymbol{x}$).

3.2 Riemannian geometry

Let $\mathcal{M}$ be a smooth manifold. A Riemannian metric on $\mathcal{M}$ is defined by a smooth 2-tensor field h. For each point $\boldsymbol{x} \in \mathcal{M}$, h assigns an inner product structure $h_{\boldsymbol{x}}$ to the corresponding tangent space $T_{\boldsymbol{x}}\mathcal{M}$ (Lee, 2018). In local coordinates $(x^1, x^2, \ldots, x^n)$, $h_{\boldsymbol{x}}$ can be represented by a positive-definite symmetric matrix $(h_{ij}(\boldsymbol{x}))_{n\times n}$. A Riemannian manifold $(\mathcal{M}, h)$ is a smooth manifold endowed with a Riemannian metric.

Given Riemannian manifolds $(\mathcal{M}, h)$ and $(\mathcal{N}, g)$, a Riemannian isometry from $(\mathcal{M}, h)$ to $(\mathcal{N}, g)$ is a diffeomorphism (a smooth bijection with its inverse smooth) $\phi : \mathcal{M} \to \mathcal{N}$ with $g_{\phi(\boldsymbol{x})}(d_{\boldsymbol{x}}\phi(\boldsymbol{u}), d_{\boldsymbol{x}}\phi(\boldsymbol{v})) = h_{\boldsymbol{x}}(\boldsymbol{u}, \boldsymbol{v})$ for all $\boldsymbol{x} \in \mathcal{M}$ and $\boldsymbol{u}, \boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}$, where $d_{\boldsymbol{x}}\phi$ is the differential of ϕ at $\boldsymbol{x}$. We say $(\mathcal{M}, h)$ and $(\mathcal{N}, g)$ are Riemannian isometric if there exists an isometry between them.

3.3 Hyperbolic geometry

Hyperbolic space is a complete, simply connected Riemannian manifold with constant negative curvature $-c, c > 0$. A hyperbolic space $\mathbb{H}^n_c$ admits several equivalent geometric models, among which the Poincaré ball model is the most widely used. The Poincaré ball model $(\mathcal{D}^n_c, h)$ consists of the open ball $\mathcal{D}^n_c = \{\boldsymbol{x} \in \mathbb{R}^n : c\|\boldsymbol{x}\|^2 < 1\}$ equipped with the Riemannian metric $h(\boldsymbol{x}) = \lambda^2_c(\boldsymbol{x})\, g^E$, where g^E is the Euclidean metric and $\lambda_c(\boldsymbol{x}) = \frac{2}{1-c\|\boldsymbol{x}\|^2}$ is the conformal factor. In standard coordinates, the metric tensor reads

$$h_{ij}(\boldsymbol{x}) = \lambda^2_c(\boldsymbol{x})\, \delta_{ij} = \frac{4\, \delta_{ij}}{(1 - c\|\boldsymbol{x}\|^2)^2}. \tag{1}$$

For $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}^n_c$, the geodesic distance is $d^{\mathcal{D}}_c(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\sqrt{c}} \cosh^{-1}\left(1 + \frac{2c\,\|\boldsymbol{x}-\boldsymbol{y}\|^2}{(1-c\|\boldsymbol{x}\|^2)(1-c\|\boldsymbol{y}\|^2)}\right)$.

4 RIEMANNIAN ISOMETRIC KERNELS

This section bridges a Riemannian isometric that imposes the hyperbolic geometry on the RKHS induced by kernel functions, thereby equipping kernel functions with an intrinsic capacity to preserve the geometry structure. Specifically, given a kernel function, we first elucidate the Riemannian metric of the submanifold induced by the canonical feature map, namely the kernel metric. Building on this, we define Riemannian isometric kernels whose canonical feature maps realize a Riemannian isometry. Then by aligning the kernel metric with hyperbolic space, we establish four different types of Riemannian isometric kernels which enjoy order preservation property on the Poincaré ball. All proofs are provided in Appendix B.

4.1 RIEMANNIAN METRIC INDUCED BY KERNEL

Let $(\mathcal{M}, h)$ be an n-dimensional Riemannian manifold and let $k : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ be a smooth PD kernel, with canonical feature map ϕ and RKHS $\mathcal{H}$. Assume ϕ is a smooth embedding; then $S = \phi(\mathcal{M})$ is a smooth submanifold of $\mathcal{H}$. For local coordinates $(x^1, \dots, x^n)$ on $\mathcal{M}$ with parametrization $\boldsymbol{\psi}$, let $\boldsymbol{\varphi} = \phi \circ \boldsymbol{\psi}$ and $\boldsymbol{z} = \phi(\boldsymbol{x})$. Then $d\boldsymbol{z} = \sum_{i=1}^{n} (\partial \boldsymbol{\varphi}/\partial x^i)\, dx^i$, and $\|d\boldsymbol{z}\|^2 = \sum_{i,j} \tilde{g}_{ij}(\boldsymbol{x})\, dx^i dx^j$, where, with $\tilde{\boldsymbol{x}} = \boldsymbol{\psi}^{-1}(\boldsymbol{x})$, $\tilde{\boldsymbol{y}} = \boldsymbol{\psi}^{-1}(\boldsymbol{y})$ and $K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) = k(\boldsymbol{\psi}(\tilde{\boldsymbol{x}}), \boldsymbol{\psi}(\tilde{\boldsymbol{y}})) = \langle \boldsymbol{\varphi}(\tilde{\boldsymbol{x}}), \boldsymbol{\varphi}(\tilde{\boldsymbol{y}}) \rangle_{\mathcal{H}}$,

$$\tilde{g}_{ij}(\boldsymbol{x}) = \Big\langle \frac{\partial \boldsymbol{\varphi}}{\partial x^i}, \frac{\partial \boldsymbol{\varphi}}{\partial x^j} \Big\rangle_{\mathcal{H}} = \frac{\partial^2 K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})}{\partial x^i\, \partial y^j}\Big|_{\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{y}} = \boldsymbol{\psi}^{-1}(\boldsymbol{x})}. \tag{2}$$

For $\boldsymbol{z} \in S$, set $g_{ij}(\boldsymbol{z}) = \tilde{g}_{ij}(\phi^{-1}(\boldsymbol{z}))$; this is well-defined since ϕ is injective. We refer to g as the **kernel metric induced by** k, thus rendering S equipped with g a Riemannian manifold (S, g).

4.2 RIEMANNIAN ISOMETRIC KERNELS ON GENERAL MANIFOLDS

By characterizing the Riemannian structure of the image of the canonical feature map, namely the submanifold $\phi(\mathcal{M})$ of the RKHS, we subsequently seek to endow this submanifold with desirable properties by imposing appropriate constraints on the kernel metric. It is natural to consider ϕ as a Riemannian isometry from $\mathcal{M}$ to $\phi(\mathcal{M})$ to carry the intrinsic geometry of $\mathcal{M}$ over the submanifold $\phi(\mathcal{M})$. In particular, we formulate this condition at a kernel level:

Definition 1 (Riemannian isometric kernel)**.** *A smooth kernel $k : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ defined on a Riemannian manifold $(\mathcal{M}, h)$ is called a* Riemannian isometric kernel *on $\mathcal{M}$ if $\phi(\mathcal{M})$ is a Riemannian manifold equipped with the kernel metric g induced by k and the canonical feature map $\phi : x \mapsto k(\cdot, x)$ is a Riemannian isometry from $(\mathcal{M}, h)$ to $(\phi(\mathcal{M}), g)$.*

We provide sufficient conditions for k to be a Riemannian isometric kernel in Theorem 1. Before starting, let us first introduce the definition of topologically embedding kernels.

Definition 2 (Topologically embedding kernel)**.** *Let (X, d) be a metric space. A kernel $k : X \times X \to \mathbb{R}$ is a topologically embedding kernel if for any $x \in X$ and any $\varepsilon > 0$, there is a $\delta > 0$, such that for any $y \in X$, we have $d(x, y) \geq \varepsilon \Longrightarrow k(x, x) + k(y, y) - 2k(x, y) \geq \delta^2$.*

It is straightforward to show that if k is a topological embedding kernel, then its canonical feature map is injective and admits a continuous inverse on its image.

Theorem 1. *If a smooth topologically embedding PD kernel $k : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ on a Riemannian manifold $(\mathcal{M}, h)$ satisfies that for each $\boldsymbol{x} \in \mathcal{M}$ and each i, j, there holds $g_{ij}(\phi(\boldsymbol{x})) = h_{ij}(\boldsymbol{x})$ where g is the kernel metric induced by k and ϕ is the canonical feature map, then $(\phi(\mathcal{M}), g)$ is a Riemannian manifold and ϕ is a Riemannian isometry from $(\mathcal{M}, h)$ to $(\phi(\mathcal{M}), g)$, indicating k is a Riemannian isometric kernel on $\mathcal{M}$.*

It follows from the preceding proof that the equation $\left\langle d_{\boldsymbol{x}}\phi(\boldsymbol{u}),\, d_{\boldsymbol{x}}\phi(\boldsymbol{v})\right\rangle_{\mathcal{H}} = h_{\boldsymbol{x}}(\boldsymbol{u},\boldsymbol{v})$ is equivalent to requiring $g_{ij}(\phi(\boldsymbol{x})) = h_{ij}(\boldsymbol{x})$ for every $\boldsymbol{x}\in\mathcal{M}$ and every i,j, which can also be written in coordinates as

$$\frac{\partial^2 K(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})}{\partial x^i\,\partial y^j}\Big|_{(\tilde{\boldsymbol{x}})=(\tilde{\boldsymbol{y}})=\boldsymbol{\psi}^{-1}(\boldsymbol{x})} = h_{ij}(\boldsymbol{x}). \tag{3}$$

We call the equation above the isometric condition. By linking the kernel directly to the Riemannian metric tensor of $\mathcal{M}$, this condition transfers the intrinsic geometry of $\mathcal{M}$ to the embedded feature submanifold in the RKHS, thereby enabling the kernel to preserve the geometric structure. Theorem 1 provides a convenient sufficient criterion for k to be a Riemannian isometric kernel: it suffices to verify that k is a smooth topological embedding kernel and satisfies the isometry condition equation 3.

4.3 Riemannian Isometric Kernels on Hyperbolic Space

Let $(\mathcal{D}_c^n, h)$ be the Poincaré ball model, and let k be a kernel defined on it. Taking the identity local parametrization, its Riemannian metric in standard coordinates is $h_{ij}(\boldsymbol{x}) = 4\,\delta_{ij}/(1-c\|\boldsymbol{x}\|^2)^2$, and the isometric condition Equation 3 simplifies to

$$\frac{\partial^2 k(\boldsymbol{x},\boldsymbol{y})}{\partial x^i\,\partial y^j}\Big|_{\boldsymbol{y}=\boldsymbol{x}} = \frac{4\,\delta_{ij}}{(1-c\|\boldsymbol{x}\|^2)^2}. \tag{4}$$

To identify kernels satisfying the isometry condition, we first design a particular solution and then extend it to a broader class of solutions.

Lemma 1. *The function $\rho:\mathcal{D}_c^n\times\mathcal{D}_c^n\to\mathbb{R}$ defined by $\rho(\boldsymbol{x},\boldsymbol{y}) = -2\|\boldsymbol{x}-\boldsymbol{y}\|^2/[(1-c\|\boldsymbol{x}\|^2)(1-c\|\boldsymbol{y}\|^2)]$ satisfies the isometric condition Equation 4. Moreover, for any smooth function f, the function $f(\rho(\boldsymbol{x},\boldsymbol{y}))$ satisfies $\left(\partial^2 f(\rho(\boldsymbol{x},\boldsymbol{y}))/\partial x^i\,\partial y^j\right)\Big|_{\boldsymbol{y}=\boldsymbol{x}} = f'(0)h_{ij}(\boldsymbol{x})$.*

Building on Lemma 1, we can construct a whole family of kernels satisfying the isometric condition.

Theorem 2. *Taking $\rho(\boldsymbol{x},\boldsymbol{y})$ defined in Lemma 1, let $k^{\text{base}}(\boldsymbol{x},\boldsymbol{y}) = \log[1-c\rho(\boldsymbol{x},\boldsymbol{y})]$ and $k_t = \exp(-tk^{\text{base}}) = (1-c\rho)^{-t}$. Define*

- *(a) (**Linear combination type**) $F_1(\boldsymbol{x},\boldsymbol{y}) = \alpha_1\sum_{m=1}^N a_m k_{t_m}(\boldsymbol{x},\boldsymbol{y})$, where $t_m>0$, $a_m>0$ for all m, N is a positive integer and $\alpha_1 = (c\sum_{m=1}^N a_m t_m)^{-1}$;*
- *(b) (**Power series type**) $F_2(\boldsymbol{x},\boldsymbol{y}) = \alpha_2\sum_{n=0}^\infty a_n k_n(\boldsymbol{x},\boldsymbol{y})$, where $a_n>0$ for all n, the convergence radius of power series $\sum_{n=0}^\infty a_n r^n$ is greater than 1. Here $\alpha_2 = (c\sum_{n=1}^\infty n a_n)^{-1}$.*
- *(c) (**Integral type**) $F_3(\boldsymbol{x},\boldsymbol{y}) = \alpha_3\int_0^\infty k_t(\boldsymbol{x},\boldsymbol{y})d\mu(t)$, where $\int_0^\infty k_t(\boldsymbol{x},\boldsymbol{y})d\mu(t)$ is smooth for $\boldsymbol{x},\boldsymbol{y}$ and μ is a positive finite measure on $(0,\infty)$ with the first moment $\int_0^\infty t d\mu(t)<\infty$ and $\alpha_3 = (c\int_0^\infty t d\mu(t))^{-1}$.*
- *(d) (**Centralization type**) $F_4(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{c}\left[-k^{\text{base}}(\boldsymbol{x},\boldsymbol{y}) + k^{\text{base}}(\boldsymbol{x},\boldsymbol{x}_0) + k^{\text{base}}(\boldsymbol{x}_0,\boldsymbol{y})\right]$, where $\boldsymbol{x}_0$ is a point in $\mathcal{D}_c^n$;*

Then all types of kernels above satisfy the isometric condition Equation 4. Moreover, they are PD and topologically embedding kernels on Poincaré ball, thus they are Riemannian isometric kernels.

We provide several specific implementations for each type:

For the linear combination type, we define the Riemannian isometric binomial kernel (RIBin),

$$k^{\text{bin}}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{ct}k_t(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{ct(1-c\rho(\boldsymbol{x},\boldsymbol{y}))^t}.$$

For the power series type, we define the Riemannian isometric exponential kernel (RIExp),

$$k^{\exp}(\boldsymbol{x},\boldsymbol{y}) = \frac{e^{-\lambda}}{c\lambda}\sum_{n=0}^{\infty}\frac{\lambda^n}{n!}(1-c\rho(\boldsymbol{x},\boldsymbol{y}))^{-n} = \frac{1}{c\lambda}\exp\left[-\lambda\left(1-\frac{1}{1-c\rho(\boldsymbol{x},\boldsymbol{y})}\right)\right].$$

For the integral type, we define the Riemannian isometric integral kernel (RIInt),

$$k^{\text{int}}(\boldsymbol{x},\boldsymbol{y}) = \int_0^{\infty}\frac{1}{c}(1-c\rho(\boldsymbol{x},\boldsymbol{y}))^{-t}e^{-t}dt = \frac{1}{c\left[1+\log\left(1-c\rho(\boldsymbol{x},\boldsymbol{y})\right)\right]}.$$

For centralization type, we construct the Riemannian isometric logarithmic kernel (RILog) with $\boldsymbol{x}_0$ in $\mathcal{D}_c^n$.

$$k^{\log}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{c}\left[-k^{\text{base}}(\boldsymbol{x},\boldsymbol{y}) + k^{\text{base}}(\boldsymbol{x},\boldsymbol{x}_0) + k^{\text{base}}(\boldsymbol{x}_0,\boldsymbol{y})\right].$$

By inheriting the Riemannian isometry property, these kernels retain the intrinsic geometry of the input manifolds, thus reducing local geometric distortion induced by the embedding.

4.4 Order preservation of Riemaannian isometric kernels

For a Riemannian isometric kernel k, its canonical map ϕ embeds the input manifold $\mathcal{M}$ into the RKHS $\mathcal{H}$. In kernel method experiments, similarity measurement is generally based on $\|\phi(x)-\phi(y)\|_{\mathcal{H}} = \sqrt{k(x,x)+k(y,y)-2k(x,y)}$. We call this the kernel-induced distance d_k of k. The relationship between the geodesic distance d_S of $S=\phi(\mathcal{M})$ and d_k is worth discussing.

Theorem 3 (Order preservation)**.** *Denote ϕ_i the canonical feature map of F_i and the corresponding submanifold $S_i=\phi_i(\mathcal{M})$. Let $\mathcal{M}=\mathcal{D}_c^n$. The kernel distance d_{F_i} of $F_i(i=1,2,3,4)$ can be expressed as $f_i(d_{S_i})$ and f_i is a monotonically increasing function. Moreover, when $\|\phi_i(\boldsymbol{x})-\phi_i(\boldsymbol{y})\|_{\mathcal{H}}\to 0$, $d_{S_i}(\boldsymbol{x},\boldsymbol{y})\to 0$.*

The theorem reveals that for any two pairs of points $(\boldsymbol{x}_1,\boldsymbol{y}_1)$ and $(\boldsymbol{x}_2,\boldsymbol{y}_2)$, if $d_k(\boldsymbol{x}_1,\boldsymbol{y}_1)<d_k(\boldsymbol{x}_2,\boldsymbol{y}_2)$, then $d_S(\boldsymbol{x}_1,\boldsymbol{y}_1)<d_S(\boldsymbol{x}_2,\boldsymbol{y}_2)$, and also implies that minimizing $\|\phi(\boldsymbol{x})-\phi(\boldsymbol{y})\|_{\mathcal{H}}$ toward zero will drive d_S to zero as well. This indicates consistency between the distance ordering in experiments and the ordering on the submanifold S.

5 Experiments

In this section, we first assess the geometric distortion of the proposed Riemannian isometric kernels via the geometric stress metric, and then evaluate them on four downstream tasks: generalized zero-shot learning (GZSL), few-shot learning (FSL), graph node classification (GNC), and semantic textual similarity (STS). Experimental settings are in the Appendix D. We compare with: 1) a no-kernel baseline (Khrulkov et al., 2020; Gao et al., 2021); 2) curvature-aware kernels (CHL, CHPoly, CHRBF, CHLap) (Yang et al., 2023); and 3) adaptive hyperbolic kernels (AHL, AHPoly, AHRBF, AHLap, AHRad) (Si et al., 2026). We also provide comparison against a series of vanilla Poincaré kernels (Fang et al., 2021; 2023) in Appendix H. All results are either from the original papers or reproduced under identical settings for fairness.

5.1 Geometric Distortion Analysis

Problem setup. We analyze geometric distortion via the stress (Kruskal, 1964) between kernel and hyperbolic embeddings, which quantifies how faithfully pairwise distances are preserved. Given distinct embeddings $\{\boldsymbol{z}_i^{(A)}\}$ and $\{\boldsymbol{z}_i^{(B)}\}$ of the same samples, stress is defined by:

$$\text{Stress}(A\to B) = \sqrt{\frac{\sum_{i<j}\left(d_c(\boldsymbol{z}_i^{(A)},\boldsymbol{z}_j^{(A)}) - d_c(\boldsymbol{z}_i^{(B)},\boldsymbol{z}_j^{(B)})\right)^2}{\sum_{i<j} d_c^2(\boldsymbol{z}_i^{(A)},\boldsymbol{z}_j^{(A)})}}. \tag{5}$$

Under hyperbolic settings, lower stress indicates better preservation of hyperbolic geometry and stronger hierarchical representation. Here we train hyperbolic prototypical networks on CIFAR-FS (Bertinetto et al., 2019) with different kernels as measures and extract corresponding embeddings to compare with the geodesic embedding ("geo"), which is the intrinsic ground-truth reference.

Experimental results. As shown in Table 1, RILog attains the lowest stress 0.0392, while RIExp, RIBin, and RIInt also outperform all other kernels. Moreover, the proposed RI kernels achieve stress values lower than other kernels (RIInt 0.0404 vs. AHRBF 0.1623, as shown in Table 1). These results sufficiently support our claim that the proposed Riemannian isometric kernels reduce geometric distortion, with RILog achieving the best geometric fidelity.

Table 1: Geometric stress of embeddings relative to the geodesic-distance reference.

	Cosine	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
$\mathrm{Stress}(\mathrm{geo}\to k)$	0.2995	0.6979	0.6999	0.6961	0.6834	0.5481	0.5506	0.1623	0.1656	0.2792	0.0399	0.0398	0.0404	**0.0392**

5.2 Zero-shot Learning

Problem setup. We use our proposed kernels as an alternative metric to evaluate their ablility of generalization under zero-shot setting. Following the kernel-learning paradigm of Fang et al. (2023), we project both the visual features and the semantic prototypes onto the Poincaré ball and replace the Euclidean compatibility score with the kernel. The baseline uses the hyperbolic geodesic distance as the similarity measure (Khrulkov et al., 2020). We evaluate on three standard GZSL benchmarks, CUB (Wah et al., 2011), AWA1 and AWA2 (Lampert et al., 2014; Xian et al., 2019). Following the standard protocol, we report the mean classification accuracy on seen (S) and unseen (U) classes together with their harmonic mean $\mathrm{H} = 2\mathrm{US}/(\mathrm{U}+\mathrm{S})$, as the primary GZSL metric.

Experimental results. Table 2 shows that the proposed RILog achieves the highest harmonic mean on all three datasets, outperforming the best adaptive kernel AHRad by 0.9%, 0.4% and 0.8% on AWA1, AWA2 and CUB respectively, demonstrating that RILog provides a more faithful geometric inductive bias for cross-modal GZSL. We also note that each Riemannian isometric kernel brings a significant improvement to the baseline network. For example, RIBin outperforms the baseline by 26.2%, 25.5% and 24.2% on AWA1, AWA2 and CUB. This further demonstrates that our Riemannian isometric kernels can embed hierarchically structured image data with consistent low distortion in the zero-shot regime.

Table 2: GZSL results (%) on CUB, AWA1 and AWA2. Best H per dataset is highlighted in **bold**.

Dataset	Metric	Baseline	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
AWA1	U	29.8	51.2	51.3	56.3	53.4	55.4	58.8	59.3	56.8	64.9	59.0	56.9	59.0	71.6
	S	76.4	84.7	85.4	82.7	88.9	85.3	85.3	85.2	82.5	83.7	83.4	84.4	84.4	76.6
	H	42.9	63.8	64.1	67.0	66.7	67.2	69.6	69.9	67.3	73.1	69.1	68.0	69.4	**74.0**
AWA2	U	25.5	44.5	42.2	45.1	44.9	49.4	46.2	48.3	47.1	69.3	50.4	51.0	50.2	73.9
	S	76.4	90.8	90.9	90.1	90.9	89.4	87.6	88.6	86.5	84.0	86.6	87.9	87.8	78.9
	H	38.2	59.8	57.6	60.1	60.1	63.6	60.5	62.5	61.0	75.9	63.7	64.6	63.9	**76.3**
CUB	U	18.6	43.3	41.7	45.0	45.2	46.2	49.0	44.7	47.4	49.0	51.0	49.9	49.5	49.1
	S	44.6	58.3	58.9	56.7	56.1	56.1	52.8	58.3	55.3	54.0	50.1	52.9	53.1	55.7
	H	26.3	49.7	48.8	50.1	50.1	50.7	50.8	50.6	51.0	51.4	50.5	51.3	51.2	**52.2**

5.3 Few-shot Learning

Problem setup. We also use our proposed kernels as an alternative metric to evaluate their ablility of recognization and generalization in few-shot case. Our experiments follow the general practice (i.e., 5-way

1-shot and 5-way 5-shot and 15 query images) to evaluate the model and adopt hyperbolic prototypical networks (Khrulkov et al., 2020) with a Conv-4 backbone following the kernel-learning paradigm of (Fang et al., 2023) which embeds the kernel into the loss in place of the original Euclidean metric (the baseline). We evaluate on CUB (Wah et al., 2011) and mini-ImageNet (Deng et al., 2009) with their standard train/validation/test splits. We adopt the mean classification accuracy as the evaluation metric and conduct experiments under the 5-way 1-shot and 5-way 5-shot settings, reporting the mean over test episodes with 95% confidence intervals.

Experimental results. As shown in Table 3, we observe that RILog attains the best accuracy on all four settings, improving the baseline by 9.0%, 6.8%, 3.6%, and 2.1% on CUB 1-shot, CUB 5-shot, mini-ImageNet 1-shot, and mini-ImageNet 5-shot, respectively, and AHRad by 4.7%, 1.8%, 1.7%, and 0.6%. RIBin, RIExp and RIInt are also consistently among the strongest methods. The largest gains over AHRad occur in the lowest-data regime, precisely where a faithful geometric inductive bias matters most, demonstrating the intrinsic geometric preservation brings about the improvement for accuracy in FSL.

Table 3: Few-shot classification accuracy (%) with 95% confidence interval on CUB and mini-ImageNet datasets. Best result for each dataset and experimental setting is highlighted in **bold**.

Dataset	Setting	Baseline	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
CUB	1-shot	$59.6_{0.2}$	$65.8_{0.2}$	$62.3_{0.2}$	$56.7_{0.2}$	$56.9_{0.2}$	$61.2_{0.2}$	$62.0_{0.2}$	$60.6_{0.2}$	$60.9_{0.2}$	$63.9_{0.2}$	$67.3_{0.2}$	$67.3_{0.2}$	$65.8_{0.2}$	**68.6**$_{0.2}$
	5-shot	$78.3_{0.2}$	$82.9_{0.2}$	$80.3_{0.2}$	$79.9_{0.2}$	$79.1_{0.2}$	$82.7_{0.1}$	$82.2_{0.2}$	$82.2_{0.2}$	$81.6_{0.2}$	$83.3_{0.2}$	**85.1**$_{0.2}$	$83.8_{0.2}$	$83.6_{0.2}$	**85.1**$_{0.2}$
mini-ImageNet	1-shot	$52.7_{0.2}$	$53.6_{0.2}$	$52.6_{0.2}$	$53.5_{0.2}$	$52.7_{0.2}$	$52.5_{0.2}$	$53.8_{0.2}$	$52.4_{0.2}$	$53.0_{0.2}$	$54.6_{0.2}$	$55.2_{0.2}$	$54.3_{0.2}$	$54.5_{0.2}$	**56.3**$_{0.2}$
	5-shot	$71.7_{0.2}$	$72.6_{0.2}$	$71.3_{0.2}$	$71.9_{0.2}$	$71.5_{0.2}$	$71.7_{0.2}$	$71.7_{0.2}$	$72.0_{0.2}$	$71.5_{0.2}$	$73.2_{0.2}$	$72.8_{0.2}$	$71.9_{0.2}$	$71.8_{0.2}$	**73.8**$_{0.2}$

5.4 Graph node classification

Problem setup. Graph node classification aims to assign labels to nodes in an attributed graph by leveraging both node features and graph structure. To verify the ability of the proposed Riemannian isometric kernels to preserve the intrinsic geometry of hyperbolic embeddings in graph node classification, we follow the kernel-learning paradigm of (Yang et al., 2023), projecting node features onto the Poincaré ball through the exponential map at the origin and performing node classification by kernel SVMs. We evaluate on five benchmark attributed graphs, Facebook, Terrorist, AC, Cora ML and LastFM, collected from the Karate Club (Rozemberczki et al., 2020) and Network Data Repository (Rossi & Ahmed, 2015) suites, with embedding dimensions $\dim \in \{2, 5, 10, 25\}$.

Experimental results. As shown in Table 4, the four Riemannian isometric kernels attain the highest accuracy on every one of the 20 dataset-dimension settings, and RILog is the single best method on 17 of them. On Cora ML, RILog outperforms the best non-RI kernel by 4.8% and 4.5% at $\dim = 10$ and $\dim = 25$, respectively. In contrast, AHRBF and AHLap degrade sharply on several graphs, dropping to 60.7% and 67.9% on Facebook at $\dim = 5$, versus 90.5% for RILog. These results show that the isometric constraint provides robustness that adaptive modulation does not and demonstrate consistent gains across diverse graph datasets and embedding dimensions.

5.5 Semantic Textual Similarity Evaluation

Problem setup. Semantic textual similarity (STS) measures the degree of semantic equivalence between two sentences and is typically evaluated by computing the correlation between model-predicted similarity scores and human-annotated judgments. We evaluate our proposed kernels in a contrastive-learning-based STS approach to assess their effectiveness in capturing semantic similarity. In doing so, we employ a pre-trained BERT-base model (Devlin et al., 2019) as the backbone and follow the supervised SimCSE paradigm (Gao et al., 2021). Specifically, we project the CLS representations into the Poincaré ball and

Table 4: Mean accuracy (%) of node classification on graph datasets. Best results are highlighted in **bold**.

Dataset	dim	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
Facebook	2	$84.3_{1.0}$	$82.0_{1.8}$	$77.5_{0.9}$	$80.1_{0.7}$	$74.1_{1.4}$	$74.4_{3.6}$	$71.9_{2.6}$	$76.0_{1.9}$	$69.4_{3.3}$	$80.6_{3.3}$	$81.8_{4.3}$	$82.9_{3.7}$	$\mathbf{85.5}_{1.2}$
	5	$89.9_{0.3}$	$89.7_{0.4}$	$88.0_{0.8}$	$88.1_{0.7}$	$86.3_{1.6}$	$83.1_{1.8}$	$60.7_{2.0}$	$67.9_{2.7}$	$79.8_{2.3}$	$87.6_{2.8}$	$88.1_{2.2}$	$88.1_{2.1}$	$\mathbf{90.5}_{0.3}$
	10	$90.6_{0.4}$	$90.6_{0.3}$	$88.4_{0.6}$	$88.8_{0.3}$	$87.0_{2.7}$	$84.3_{2.5}$	$59.0_{2.8}$	$66.8_{3.3}$	$81.4_{2.2}$	$88.0_{1.7}$	$89.5_{1.3}$	$89.6_{1.4}$	$\mathbf{91.4}_{0.8}$
	25	$90.8_{0.2}$	$90.9_{0.3}$	$88.4_{0.6}$	$88.6_{0.3}$	$87.2_{1.9}$	$85.1_{1.7}$	$58.9_{3.2}$	$64.4_{3.2}$	$83.1_{2.1}$	$88.3_{2.1}$	$90.1_{1.1}$	$90.2_{1.5}$	$\mathbf{91.5}_{0.4}$
Terrorist	2	$83.1_{2.5}$	$81.1_{1.4}$	$80.8_{2.3}$	$82.5_{2.3}$	$83.1_{3.1}$	$82.8_{2.0}$	$82.3_{3.1}$	$82.6_{3.6}$	$82.0_{2.6}$	$85.6_{2.4}$	$86.3_{3.1}$	$\mathbf{87.3}_{1.9}$	$86.4_{1.9}$
	5	$89.3_{0.9}$	$86.9_{0.5}$	$77.3_{2.2}$	$83.6_{2.1}$	$87.0_{2.1}$	$87.0_{1.8}$	$79.6_{1.0}$	$85.8_{1.5}$	$89.3_{2.8}$	$88.3_{3.2}$	$89.2_{2.2}$	$87.8_{1.3}$	$\mathbf{90.9}_{2.5}$
	10	$88.6_{1.2}$	$86.1_{2.6}$	$75.1_{2.1}$	$83.0_{1.8}$	$89.5_{1.7}$	$87.2_{1.8}$	$78.2_{2.5}$	$85.6_{2.9}$	$87.8_{2.6}$	$88.9_{2.3}$	$88.9_{2.5}$	$89.2_{2.5}$	$\mathbf{91.5}_{2.0}$
	25	$89.5_{2.3}$	$86.1_{1.8}$	$75.6_{1.6}$	$83.5_{3.2}$	$89.3_{2.0}$	$86.9_{1.6}$	$77.3_{2.6}$	$85.2_{2.6}$	$87.9_{2.1}$	$88.9_{2.0}$	$89.8_{2.7}$	$88.7_{2.3}$	$\mathbf{91.6}_{1.5}$
AC	2	$76.3_{0.9}$	$75.4_{1.1}$	$75.7_{0.8}$	$77.3_{0.8}$	$74.9_{1.5}$	$76.3_{1.4}$	$72.8_{1.4}$	$76.6_{1.2}$	$74.7_{2.5}$	$77.4_{0.5}$	$76.5_{0.8}$	$75.6_{0.7}$	$\mathbf{79.8}_{1.1}$
	5	$86.3_{0.8}$	$85.9_{0.6}$	$85.5_{0.8}$	$85.1_{1.1}$	$84.4_{0.9}$	$80.9_{0.5}$	$58.6_{3.4}$	$71.2_{1.5}$	$80.3_{1.0}$	$84.9_{1.5}$	$86.4_{1.5}$	$85.9_{1.4}$	$\mathbf{88.2}_{1.6}$
	10	$\mathbf{89.5}_{0.6}$	$88.8_{0.5}$	$87.6_{0.8}$	$88.1_{0.8}$	$85.9_{1.3}$	$84.0_{1.0}$	$58.9_{3.1}$	$72.3_{1.8}$	$83.3_{1.1}$	$88.4_{0.6}$	$88.2_{0.8}$	$87.7_{0.8}$	$\mathbf{89.5}_{1.8}$
	25	$89.6_{0.5}$	$89.3_{0.4}$	$87.5_{0.5}$	$87.8_{0.7}$	$85.8_{1.0}$	$84.7_{0.6}$	$57.4_{3.1}$	$71.0_{2.0}$	$83.9_{0.3}$	$89.4_{0.9}$	$89.3_{0.9}$	$89.3_{1.6}$	$\mathbf{90.5}_{0.6}$
Cora ML	2	$71.4_{1.6}$	$70.4_{1.5}$	$69.7_{1.1}$	$71.7_{1.2}$	$71.3_{1.5}$	$70.8_{1.7}$	$70.3_{1.5}$	$72.3_{1.6}$	$61.3_{1.2}$	$73.7_{3.3}$	$74.3_{2.7}$	$73.6_{2.2}$	$\mathbf{74.8}_{1.5}$
	5	$82.2_{2.2}$	$79.7_{1.4}$	$48.8_{1.0}$	$61.0_{1.2}$	$82.2_{2.2}$	$79.7_{1.4}$	$52.8_{1.2}$	$64.9_{1.1}$	$77.1_{0.9}$	$79.7_{1.8}$	$80.4_{1.0}$	$80.6_{1.0}$	$\mathbf{83.3}_{2.7}$
	10	$83.9_{0.8}$	$82.0_{0.7}$	$46.1_{1.1}$	$59.9_{2.1}$	$83.9_{0.8}$	$82.0_{0.7}$	$50.0_{0.8}$	$63.2_{1.3}$	$80.6_{1.2}$	$83.9_{1.3}$	$83.2_{1.6}$	$84.1_{1.1}$	$\mathbf{88.7}_{1.8}$
	25	$84.8_{1.1}$	$81.9_{1.1}$	$44.8_{1.5}$	$58.3_{1.2}$	$84.8_{1.1}$	$82.1_{1.1}$	$48.5_{1.5}$	$61.9_{1.1}$	$80.3_{0.8}$	$84.6_{0.8}$	$84.7_{0.8}$	$84.7_{1.0}$	$\mathbf{89.3}_{1.6}$
LastFM	2	$85.9_{2.9}$	$81.9_{3.0}$	$74.4_{2.9}$	$82.9_{3.5}$	$85.5_{3.9}$	$81.7_{3.3}$	$76.0_{2.8}$	$83.9_{3.0}$	$81.7_{3.0}$	$85.9_{3.7}$	$85.5_{4.0}$	$86.3_{3.3}$	$\mathbf{88.0}_{1.8}$
	5	$90.2_{2.3}$	$86.8_{1.6}$	$46.6_{2.3}$	$63.0_{2.8}$	$90.2_{2.3}$	$86.9_{1.8}$	$51.1_{2.5}$	$65.9_{3.9}$	$86.3_{1.5}$	$90.2_{3.6}$	$90.2_{3.6}$	$\mathbf{90.7}_{3.7}$	$90.3_{0.8}$
	10	$90.5_{1.8}$	$88.9_{2.1}$	$47.8_{2.8}$	$61.3_{3.2}$	$90.5_{1.8}$	$88.8_{1.7}$	$51.5_{3.1}$	$64.2_{2.7}$	$87.1_{1.8}$	$91.7_{2.4}$	$91.4_{2.3}$	$91.5_{2.2}$	$\mathbf{91.9}_{1.2}$
	25	$90.8_{1.7}$	$89.2_{1.3}$	$45.6_{2.5}$	$62.2_{4.6}$	$90.8_{1.7}$	$88.9_{1.3}$	$51.1_{3.0}$	$65.5_{4.4}$	$86.9_{0.5}$	$92.1_{2.4}$	$\mathbf{92.4}_{2.2}$	$\mathbf{92.4}_{2.5}$	$92.3_{1.7}$

replace the cosine similarity used in the original SimCSE objective function with the proposed kernel-based similarity. We adopt the same comparison methods as those used in the zero-shot learning experiments. The original SimCSE, which uses cosine similarity to measure the similarity between sentence representations, serves as the baseline. For supervised pre-training, we use a combined dataset constructed from MNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015), while the evaluation is conducted on the STS-B benchmark (Cer et al., 2017). Following the standard STS evaluation protocol, we report Spearman's rank correlation (SRC) between the similarity scores predicted by the model and the corresponding human-annotated ground-truth scores.

Experimental results. As shown in Table 5, RILog achieves the highest Spearman correlation value of 86.31, outperforming the cosine-similarity baseline by 2.07% and the best-performing adaptive kernel, AHRad, by 1.15%. RIBin, RIExp, and RIInt also consistently outperform all Poincaré and curvature-aware kernels, demonstrating the effectiveness of the proposed Riemannian isometric kernels for learning textual representations and further highlighting their generalizability beyond visual tasks.

Table 5: Semantic textual similarity results on STS-B benchmark. The best result is highlighted in **bold**.

Method	Baseline	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
SRC	84.24	84.70	84.63	83.98	84.74	84.27	84.68	84.33	84.48	85.16	84.92	85.53	85.45	**86.31**

6 Conclusion

In this paper, we alleviated the geometric distortion problem in hyperbolic kernel methods by shifting the focus from metric-space isometry to Riemannian isometry at the manifold level. We introduced the notion of Riemannian isometric kernels, proved a criterion for identifying them, and constructed four concrete families on the Poincaré ball, including RILog, RIBin, RIExp, and RIInt. We proved their positive definiteness and order-preservation properties. Extensive experiments on geometric stress, GZSL, few-shot learning, graph node classification, and semantic textual similarity demonstrated that our kernels outperform existing state-of-the-art hyperbolic kernels, especially in low-data regimes where faithful geometric inductive biases are critical. Our work provides a new method for designing kernel methods that rigorously preserve the intrinsic geometry of data manifolds.

AI USE STATEMENT

We used generative AI tools solely for language polishing, including grammar, wording, and readability improvements of this manuscript. We did not use them for research ideation, literature review, data collection or analysis, code generation, experimental design, or generating scientific claims. All AI-assisted edits were reviewed and verified by the authors, and we take full responsibility for the final content of this work.

REFERENCES

Shun-ichi Amari and Si Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 1999.

Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Springer-Verlag, 1984.

Luca Bertinetto, João F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *Proceedings of the International Conference on Learning Representations*, 2019.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *Proceedings of the International Conference on Learning Representations*, 2018.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher David Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, 2017.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Hyunghoon Cho, Benjamin DeMeo, Jian Peng, and Bonnie Berger. Large-margin classification in hyperbolic space. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

Pengfei Fang, Mehrtash Harandi, and Lars Petersson. Kernel methods in hyperbolic spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Pengfei Fang, Mehrtash Harandi, Zhenzhong Lan, and Lars Petersson. Poincaré kernels for hyperbolic representations. *International Journal of Computer Vision*, 2023.

Jacques Faraut and Khelifa Harzallah. Distances hilbertiennes invariantes sur un espace homogène. *Annales de l'Institut Fourier*, 1974.

Aasa Feragen, François Lauze, and Søren Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Matthias Hamann. On the tree-likeness of hyperbolic spaces. *Mathematical Proceedings of the Cambridge Philosophical Society*, 2018.

Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on riemannian manifolds with gaussian RBF kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Joseph Bernard Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 1964.

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*, 2014.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

John M. Lee. *Introduction to Riemannian Manifolds*. Springer International Publishing, 2018.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, 2017.

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate club: An API oriented open-source Python framework for unsupervised learning on graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *Graph Drawing*, 2012.

Isaac Jacob Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 1938.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv*, 2018.

Leping Si, Meimei Yang, Hui Xue, Shipeng Zhu, and Pengfei Fang. Adaptive hyperbolic kernels: Modulated embedding in de branges-rovnyak spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer New York, 2008.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. Caltech-UCSD birds-200-2011. Technical report, California Institute of Technology, 2011.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 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*, 2019.

Meimei Yang, Pengfei Fang, and Hui Xue. Expanding the hyperbolic kernels: A curvature-aware isometric embedding view. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023.

Bin Zhao, Prithviraj Sen, and Lise Getoor. Entity and relationship labeling in affiliation networks. In *Proceedings of the International Conference on Machine Learning Workshop on Statistical Network Analysis*, 2006.

A POSITIVE DEFINITE KERNELS AND NEGATIVE DEFINITE KERNELS

In this section, we recall the standard definitions of positive definite, negative definite, and conditional positive definite kernels over the real field, together with several elementary closure properties that are used in the proof of Theorem 2. Throughout, let $\mathcal{X}$ be a nonempty set.

Definition A.1 (Positive definite kernel (Berg et al., 1984)). *A symmetric kernel $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called* positive definite *(PD) if for every $n \in \mathbb{N}$, every choice of points $x_1, \dots, x_n \in \mathcal{X}$, and every choice of real coefficients $c_1, \dots, c_n \in \mathbb{R}$, we have*

$$\sum_{i,j=1}^{n} c_i c_j \, \psi(x_i, x_j) \geq 0.$$

If the inequality is strict for all nonzero vectors (c_i), the kernel is called strictly positive definite.

Definition A.2 (Negative definite kernel and conditional positive kernel (Berg et al., 1984)). *A symmetric kernel $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called* negative definite *(nd) if for every $n \in \mathbb{N}$, every choice of points $x_1, \dots, x_n \in \mathcal{X}$, and every choice of real coefficients $c_1, \dots, c_n$ with $\sum_{i=1}^{n} c_i = 0$, we have*

$$\sum_{i,j=1}^{n} c_i c_j \, \psi(x_i, x_j) \leq 0.$$

Proposition A.1 (Closure properties of positive definite kernels (Berg et al., 1984)). *The following closure properties hold for kernels on a fixed nonempty set $\mathcal{X}$:*

(a) ***Addition and positive scaling.*** *The sum of two PD kernels is PD; a positive scalar multiple of a PD kernel is PD.*

(b) ***Multiplication (Schur product).*** *If k_1 and k_2 are positive definite, then their pointwise product $k_1 k_2$ is positive definite.*

(c) ***Nonnegative power series.*** *Let $(a_n)_{n=0}^{\infty} \subset \mathbb{R}_{\geq 0}$ such that $\sum_{n=0}^{\infty} a_n r^n$ converges for all $r \in [0, 1]$. If k is PD with $|k(x, y)| \leq 1$ for all $x, y \in \mathcal{X}$, then*

$$K(x, y) = \sum_{n=0}^{\infty} a_n \big(k(x, y)\big)^n$$

is PD.

(d) ***Pointwise limits.*** *If $(k_m)_{m=1}^{\infty}$ is a sequence of PD kernels and $k(x, y) = \lim_{m\to\infty} k_m(x, y)$ exists pointwise for all $x, y \in \mathcal{X}$, then k is PD.*

(e) ***Integration against a nonnegative measure.*** *Let $(\mathcal{T}, \mu)$ be a measure space with $\mu \geq 0$, and let k_t be a family of PD kernels such that*

$$K(x, y) = \int_{\mathcal{T}} k_t(x, y) \, d\mu(t)$$

exists and is finite for each $x, y \in \mathcal{X}$. Then K is PD.

Theorem A.1 (Berg et al. (1984)). *Let $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric kernel and $x_0 \in \mathcal{X}$. Put $\varphi(x, y) = \psi(x, x_0) + \psi(x_0, y) - \psi(x_0, x_0) - \psi(x, y)$. Then φ is PD if and only if ψ is nd.*

Theorem A.2 (Berg et al. (1984)). *Let $\psi : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric kernel. Then ψ is negative definite if and only if for every $t > 0$, the kernel*

$$K_t(x, y) = \exp\big(-t \, \psi(x, y)\big)$$

is positive definite. Equivalently, ψ is negative definite if and only if $\exp(-t\psi)$ is positive definite for all $t > 0$.

B COMPLETE PROOF OF LEMMAS AND THEOREMS

B.1 COMPLETE PROOF OF THEOREM 1

Proof. We first prove that ϕ is a homeomorphism between $\mathcal{M}$ and $\phi(\mathcal{M})$. Note that k is a topologically embedding kernel, ϕ is injective and ϕ^{-1} is continuous. Besides, given the smoothness of k, ϕ is also smooth (Steinwart & Christmann, 2008), which implies ϕ is a homeomorphism between $\mathcal{M}$ and $\phi(\mathcal{M})$.

Using the same notation as in section 4.1, for every $\boldsymbol{x} \in \mathcal{M}$ and every pair of tangent vectors $\boldsymbol{u}, \boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}$, we have

$$\begin{aligned}\left\langle d_x\phi(\boldsymbol{u}),\, d_x\phi(\boldsymbol{v})\right\rangle_{\mathcal{H}} &= \sum_{i,j}\left\langle u^i\frac{\partial \boldsymbol{\varphi}}{\partial x^i}, v^j\frac{\partial \boldsymbol{\varphi}}{\partial x^j}\right\rangle_{\mathcal{H}} = \sum_{i,j} u^i v^j \frac{\partial^2 K(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})}{\partial x^i\,\partial y^j}\Big|_{\tilde{\boldsymbol{x}}=\tilde{\boldsymbol{y}}=\boldsymbol{\psi}^{-1}(\boldsymbol{x})} \\ &= \sum_{i,j} u^i v^j \tilde{g}_{ij}(\boldsymbol{x}) = \sum_{i,j} u^i v^j g_{ij}(\phi(\boldsymbol{x})) = \sum_{i,j} u^i v^j h_{ij}(\boldsymbol{x}).\end{aligned}$$

This means that for any $\boldsymbol{u}, \boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}$, $\left\langle d_{\boldsymbol{x}}\phi(\boldsymbol{u}),\, d_{\boldsymbol{x}}\phi(\boldsymbol{v})\right\rangle_{\mathcal{H}} = h_{\boldsymbol{x}}(\boldsymbol{u},\boldsymbol{v})$, where the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ is taken as the Riemannian metric on $\mathcal{H}$. Thus, $d_{\boldsymbol{x}}\phi$ is injective for each point since

$$d_x\phi(\boldsymbol{u}) = 0 \Longrightarrow \left\langle d_{\boldsymbol{x}}\phi(\boldsymbol{u}),\, d_{\boldsymbol{x}}\phi(\boldsymbol{u})\right\rangle_{\mathcal{H}} = 0 \Longrightarrow h_{\boldsymbol{x}}(\boldsymbol{u},\boldsymbol{u}) = 0 \Longrightarrow \boldsymbol{u} = 0.$$

It follows immediately that ϕ is a smooth embedding. Noting that ϕ is a bijection form $\mathcal{M}$ to $\phi(\mathcal{M})$, by the corollary of global rank theorem (Lee, 2018), ϕ is a diffeomorphism from $\mathcal{M}$ to $\phi(\mathcal{M})$. This makes $\phi(\mathcal{M})$ a smooth manifold.

Subsequently, equipping it with the metric g, $(\phi(\mathcal{M}), g)$ is a Riemannian manifold. Since for any $\boldsymbol{u}, \boldsymbol{v} \in T_{\boldsymbol{x}}\mathcal{M}$, $\left\langle d_{\boldsymbol{x}}\phi(\boldsymbol{u}),\, d_{\boldsymbol{x}}\phi(\boldsymbol{v})\right\rangle_{\mathcal{H}} = h_{\boldsymbol{x}}(\boldsymbol{u},\boldsymbol{v})$, by definition, ϕ is a Riemannian isometry from $(\mathcal{M}, h)$ to $(\phi(\mathcal{M}), g)$. □

B.2 COMPLETE PROOF OF LEMMA 1

We first compute the derivative of ρ with respect to y^j:

$$\frac{\partial \rho(\boldsymbol{x},\boldsymbol{y})}{\partial y^j} = -\frac{4\big[(y_j - x_j)(1 - c\|\boldsymbol{y}\|^2) + c\,y_j\|\boldsymbol{x}-\boldsymbol{y}\|^2\big]}{(1-c\|\boldsymbol{x}\|^2)(1-c\|\boldsymbol{y}\|^2)^2}. \tag{6}$$

Differentiating once more with respect to x^i yields

$$\begin{aligned}\frac{\partial^2 \rho(\boldsymbol{x},\boldsymbol{y})}{\partial x^i\,\partial y^j} =& \frac{4\big[\delta_{ij}(1-c\|\boldsymbol{y}\|^2) - 2c\,y_j(x_i - y_i)\big](1-c\|\boldsymbol{x}\|^2)}{(1-c\|\boldsymbol{x}\|^2)^2(1-c\|\boldsymbol{y}\|^2)^2} \\ &+ \frac{4c\,x_i\big[(x_j - y_j)(1-c\|\boldsymbol{y}\|^2) - c\,y_j\|\boldsymbol{x}-\boldsymbol{y}\|^2\big]}{(1-c\|\boldsymbol{x}\|^2)^2(1-c\|\boldsymbol{y}\|^2)^2}.\end{aligned}$$

Substituting $\boldsymbol{y} = \boldsymbol{x}$ into the above expression, we obtain

$$\frac{\partial^2 \rho(\boldsymbol{x},\boldsymbol{y})}{\partial x^i\,\partial y^j}\Big|_{\boldsymbol{y}=\boldsymbol{x}} = \frac{4\,\delta_{ij}}{(1-c\|\boldsymbol{x}\|^2)^2}, \tag{7}$$

which is exactly the isometric condition Equation 4. Therefore, ρ satisfies the isometric condition equation.

Then we perform a direct computation for $f(\rho(\boldsymbol{x},\boldsymbol{y}))$ where f is an arbitrary smooth function.

$$\frac{\partial}{\partial x^i}\frac{\partial f(\rho)}{\partial y^j} = \frac{\partial}{\partial x^i}\left(f'(\rho)\frac{\partial \rho}{\partial y^j}\right) = \frac{\partial f'(\rho)}{\partial x^i}\frac{\partial \rho}{\partial y^j} + f'(\rho)\frac{\partial^2 \rho}{\partial x^i \partial y^j}. \tag{8}$$

By the Equation 6, we have $\partial\rho/\partial y^j|_{\boldsymbol{y}=\boldsymbol{x}} = 0$. And note that $\rho(\boldsymbol{x},\boldsymbol{x}) = 0$, we have

$$\frac{\partial^2 f(\rho(\boldsymbol{x},\boldsymbol{y}))}{\partial x^i \partial y^j}\Big|_{\boldsymbol{y}=\boldsymbol{x}} = f'(\rho(\boldsymbol{x},\boldsymbol{x})) \cdot \frac{\partial^2 \rho(\boldsymbol{x},\boldsymbol{y})}{\partial x^i \partial y^j}\Big|_{\boldsymbol{y}=\boldsymbol{x}} = f'(0) h_{ij}(\boldsymbol{x}). \tag{9}$$

This completes the proof. □

B.3 Complete proof of Theorem 2

Before we prove the theorem, we introduce another model of hyperbolic space, called the hyperboloid model or Lorentz model, which is helpful for our proof.

Let $\mathbb{R}^{n,1}$ be the $(n+1)$-dimensional Minkowski space equipped with the Lorentz inner product $\langle x, y\rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^n x_i y_i$. The hyperboloid sheet is

$$\mathcal{L}_c^n = \big\{x = (x_0, \ldots, x_n) \in \mathbb{R}^{n,1} : \langle x, x\rangle_{\mathcal{L}} = -\tfrac{1}{c},\ x_0 > 0\big\}, \tag{10}$$

with geodesic distance $d_c^{\mathcal{L}}(x,y) = \frac{1}{\sqrt{c}} \cosh^{-1}\big(-c\langle x, y\rangle_{\mathcal{L}}\big)$.

The Poincaré ball model and the hyperboloid model are related by the isometric diffeomorphism $\Pi : \mathcal{D}_c^n \to \mathcal{L}_c^n$ given by

$$\Pi(z) = \Big(\lambda_c(z)\,\frac{1 + c\|z\|^2}{2\sqrt{c}},\ \lambda_c(z)\,z\Big), \qquad \Pi^{-1}(x) = \frac{(x_1, \ldots, x_n)}{1 + \sqrt{c}\,x_0}, \tag{11}$$

where $x = (x_0, x_1, \ldots, x_n) \in \mathcal{L}_c^n$.

Then we start our proof.

Proof. First, for $F_4(\boldsymbol{x},\boldsymbol{y})$, note that

$$\frac{\partial^2 F_4(\boldsymbol{x},\boldsymbol{y})}{\partial x^i \partial y^j} = \frac{\partial^2}{\partial x^i \partial y^j}\left(\frac{1}{c}\left(-k^{\text{base}}(\boldsymbol{x},\boldsymbol{y}) + k^{\text{base}}(\boldsymbol{x},\boldsymbol{x}_0) + k^{\text{base}}(\boldsymbol{x}_0,\boldsymbol{y})\right)\right) = -\frac{1}{c}\frac{\partial^2 k^{\text{base}}}{\partial x^i \partial y^j}.$$

Since

$$\frac{d}{d\rho}\left(\log(1 - c\rho)\right)\Big|_{\rho=0} = \left(\frac{-c}{1 - c\rho}\right)\Big|_{\rho=0} = -c, \tag{12}$$

we have

$$\frac{\partial^2 k^{\text{base}}(\boldsymbol{x},\boldsymbol{y})}{\partial x^i \partial y^j}\Big|_{\boldsymbol{y}=\boldsymbol{x}} = -c h_{ij}(\boldsymbol{x}). \tag{13}$$

Therefore, we get

$$\frac{\partial^2 F_4(\boldsymbol{x},\boldsymbol{y})}{\partial x^i \partial y^j}\Big|_{\boldsymbol{y}=\boldsymbol{x}} = -\frac{1}{c}\frac{\partial^2 k^{\text{base}}(\boldsymbol{x},\boldsymbol{y})}{\partial x^i \partial y^j}\Big|_{\boldsymbol{y}=\boldsymbol{x}} = h_{ij}(\boldsymbol{x}) \tag{14}$$

To prove the other three types of kernels satisfy the isometric condition Equation 4, noting that they can all be expressed as composite functions of ρ, we just need to verify that the derivative value of the outer function of each type of kernel at 0 is 1.

For $k_t(\boldsymbol{x},\boldsymbol{y}) = (1 - c\rho)^{-t}$, we have

$$\frac{d}{d\rho}\left(\frac{1}{(1 - c\rho)^t}\right)\Big|_{\rho=0} = \frac{ct}{(1 - c\rho)^{t+1}}\Big|_{\rho=0} = ct.$$

Denoting $g_t(\rho) = (1 - c\rho)^{-t}$, we know $g_t'(0) = ct$ for any $t > 0$. So we have

$$\frac{d}{d\rho}\alpha_1 \sum_{m=1}^{N} a_m g_{t_m}(\rho)\Big|_{\rho=0} = \alpha_1 \sum_{m=1}^{N} a_m g_{t_m}'(0) = \alpha_1 c \sum_{m=1}^{N} a_m t_m = 1,$$
$$\frac{d}{d\rho}\alpha_2 \sum_{n=0}^{\infty} a_n g_n(\rho)\Big|_{\rho=0} = \alpha_2 \sum_{n=0}^{\infty} a_n g_n'(0) = \alpha_2 c \sum_{n=0}^{\infty} n a_n = 1,$$
$$\frac{d}{d\rho}\alpha_3 \int_0^{\infty} g_t(\rho) d\mu(t)\Big|_{\rho=0} = \alpha_3 \int_0^{\infty} g_t'(0) d\mu(t) = \alpha_3 c \int_0^{\infty} t d\mu(t) = 1.$$

Thus, we have proved that all types of kernels satisfy the isometric condition Equation 4. The smoothness of kernels is obvious because the composite function of a smooth function is still a smooth function. Therefore, it is only necessary to prove that these kernels are topologically embedding kernels. The distance on Poincaré ball is $d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\sqrt{c}} \cosh^{-1}(1 - c\rho(\boldsymbol{x}, \boldsymbol{y}))$, so for any $\varepsilon > 0$, if $d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}) \geq \varepsilon$, since $\cosh^{-1}(r)$ is monotonically increasing, we have

$$1 - c\rho \geq \cosh(\sqrt{c} \cdot \varepsilon) > 1.$$

Then for $F_4(\boldsymbol{x}, \boldsymbol{y})$, we have

$$\begin{aligned} F_4(\boldsymbol{x}, \boldsymbol{x}) + F_4(\boldsymbol{y}, \boldsymbol{y}) - 2F_4(\boldsymbol{x}, \boldsymbol{y}) =& \frac{2}{c} \log[1 - c\rho(\boldsymbol{x}, \boldsymbol{y})] \\ \geq& \frac{2}{c} \log \cosh(\sqrt{c} \cdot \varepsilon) =: \delta^{\log} > 0. \end{aligned}$$

So F_4 is a topologically embedding kernel. Then for F_1, F_2 and F_3, since $k_t(\boldsymbol{x}, \boldsymbol{x}) = 1$, we have

$$\begin{aligned} F_1(\boldsymbol{x}, \boldsymbol{x}) + F_1(\boldsymbol{y}, \boldsymbol{y}) - 2F_1(\boldsymbol{x}, \boldsymbol{y}) =& 2\alpha_1 \sum_{m=1}^{N} a_m - 2\alpha_1 \sum_{m=1}^{N} a_m (1 - c\rho(\boldsymbol{x}, \boldsymbol{y}))^{-t_m} \\ \geq& 2\alpha_1 \sum_{m=1}^{N} a_m \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot \varepsilon)^{t_m}}\right) =: \delta_1 > 0, \\ F_2(\boldsymbol{x}, \boldsymbol{x}) + F_2(\boldsymbol{y}, \boldsymbol{y}) - 2F_2(\boldsymbol{x}, \boldsymbol{y}) =& 2\alpha_2 \sum_{n=0}^{\infty} a_n - 2\alpha_2 \sum_{n=0}^{\infty} a_n (1 - c\rho(\boldsymbol{x}, \boldsymbol{y}))^{-n} \\ \geq& 2\alpha_2 \sum_{n=0}^{\infty} a_n \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot \varepsilon)^{n}}\right) =: \delta_2 > 0, \\ F_3(\boldsymbol{x}, \boldsymbol{x}) + F_3(\boldsymbol{y}, \boldsymbol{y}) - 2F_3(\boldsymbol{x}, \boldsymbol{y}) =& 2\alpha_3 \int_0^{\infty} d\mu(t) - 2\alpha_3 \int_0^{\infty} (1 - c\rho(\boldsymbol{x}, \boldsymbol{y}))^{-t} d\mu(t), \\ \geq& 2\alpha_3 \int_0^{\infty} \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot \varepsilon)^{t}}\right) d\mu(t) =: \delta_3 > 0. \end{aligned}$$

Here δ_2 and δ_3 are well-defined since $\sum_{n=0}^{\infty} a_n r^n$ converges at $r = 1$ and $\mu(t)$ is a finite measure, which implies that

$$\sum_{n=0}^{\infty} a_n \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot \varepsilon)^{n}}\right) \leq \sum_{n=0}^{\infty} a_n < \infty,$$
$$\int_0^{\infty} \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot \varepsilon)^{t}}\right) d\mu(t) \leq \int_0^{\infty} d\mu(t) < \infty.$$

Thus, we prove that F_4, F_1, F_2 and F_3 are topologically embedding kernels. Then we prove they are PD.

For $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}_c^n$, take the isometric diffeomorphism $\Pi : \mathcal{D}_c^n \to \mathcal{L}_c^n$, where $\mathcal{L}_c^n$ is the hyperboloid model of curvature $-c$. Since Π is an isometric diffeomorphism, $d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}) = d_c^{\mathcal{L}}(\Pi(\boldsymbol{x}), \Pi(\boldsymbol{y}))$, and by the monotonicity of $\cosh^{-1}$ we have $1 - c\rho(\boldsymbol{x}, \boldsymbol{y}) = -c\,\langle \Pi(\boldsymbol{x}), \Pi(\boldsymbol{y})\rangle_{\mathcal{L}}$. Therefore,

$$k^{\text{base}}(\boldsymbol{x}, \boldsymbol{y}) = \log\big[-c\,\langle \Pi(\boldsymbol{x}), \Pi(\boldsymbol{y})\rangle_{\mathcal{L}}\big]. \tag{15}$$

By a classical result of Faraut & Harzallah (1974), $\log(-c\langle \boldsymbol{z}, \boldsymbol{w}\rangle_{\mathcal{L}})$ is a negative-definite kernel on $\mathcal{L}_c^n$, i.e. for any $c_i, c_j \in \mathbb{R}$ with $\sum_i c_i = 0$ and arbitrary $\boldsymbol{z}_i \in \mathcal{L}_c^n$,

$$\sum_{i,j} c_i c_j \log\big(-c\langle \boldsymbol{z}_i, \boldsymbol{z}_j\rangle_{\mathcal{L}}\big) \le 0. \tag{16}$$

Consequently, for arbitrary $\boldsymbol{x}_i \in \mathcal{D}_c^n$,

$$\sum_{i,j} c_i c_j \Big[\log\big(-c\langle \Pi(\boldsymbol{x}_i), \Pi(\boldsymbol{x}_j)\rangle_{\mathcal{L}}\big)\Big] \le 0, \tag{17}$$

from which the negative definiteness of $k^{\text{base}}(\boldsymbol{x}, \boldsymbol{y})$ follows immediately.

Therefore, by Theorem A.1 and $k^{\text{base}}(\boldsymbol{x}, \boldsymbol{x}) = 0$, F_4 is PD. Then according to Theorem A.2, for any $t > 0$, the kernel

$$k_t(\boldsymbol{x}, \boldsymbol{y}) = \exp(-tk^{\text{base}}(\boldsymbol{x}, \boldsymbol{y})) = \exp(-t\log(1 - c\rho(\boldsymbol{x}, \boldsymbol{y}))) = (1 - c\rho(\boldsymbol{x}, \boldsymbol{y}))^{-t}$$

is PD. Since F_1 is the non-negative linear combination of k_t, it is PD. And note that $0 \le k_t \le 1$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}_c^n$ and

$$F_2(\boldsymbol{x}, \boldsymbol{y}) = \alpha_2 \sum_{n=0}^{\infty} a_n k_1^n(\boldsymbol{x}, \boldsymbol{y}), \tag{18}$$

$$F_3(\boldsymbol{x}, \boldsymbol{y}) = \alpha_3 \int_0^{\infty} k_t(\boldsymbol{x}, \boldsymbol{y}) d\mu(t), \tag{19}$$

so by Proposition A.1, F_2, F_3 are PD.

Since $F_i(1 = 1, 2, 3, 4)$ satisfy the isometric condition Equation 4 and they are topologically embedding PD kernels, so by 1, they are Riemannian isometric kernels. □

B.4 Complete proof of Theorem 3

By the property of Riemannian isometry, $d_{S_i}(\phi_i(\boldsymbol{x}), \phi_i(\boldsymbol{y})) = d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y})$. According to the proof process of 2, we have

$$
\begin{aligned}
d_{F_1}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{F_1(\boldsymbol{x}, \boldsymbol{x}) + F_1(\boldsymbol{y}, \boldsymbol{y}) - 2F_1(\boldsymbol{x}, \boldsymbol{y})} &= \sqrt{2\alpha_1 \sum_{m=1}^{N} a_m \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}))^{t_m}}\right)} \\
&= \sqrt{2\alpha_1 \sum_{m=1}^{N} a_m \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot d_{S_1}(\boldsymbol{x}, \boldsymbol{y}))^{t_m}}\right)}, \\
d_{F_2}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{F_2(\boldsymbol{x}, \boldsymbol{x}) + F_2(\boldsymbol{y}, \boldsymbol{y}) - 2F_2(\boldsymbol{x}, \boldsymbol{y})} &= \sqrt{2\alpha_2 \sum_{n=0}^{\infty} a_n \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}))^{n}}\right)} \\
&= \sqrt{2\alpha_2 \sum_{n=0}^{\infty} a_n \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot d_{S_2}(\boldsymbol{x}, \boldsymbol{y}))^{n}}\right)}, \\
d_{F_3}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{F_3(\boldsymbol{x}, \boldsymbol{x}) + F_3(\boldsymbol{y}, \boldsymbol{y}) - 2F_3(\boldsymbol{x}, \boldsymbol{y})} &= \sqrt{2\alpha_3 \int_0^{\infty} \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}))^{t}}\right) d\mu(t)} \\
&= \sqrt{2\alpha_3 \int_0^{\infty} \left(1 - \frac{1}{\cosh(\sqrt{c} \cdot d_{S_3}(\boldsymbol{x}, \boldsymbol{y}))^{t}}\right) d\mu(t)}, \\
d_{F_4}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{F_4(\boldsymbol{x}, \boldsymbol{x}) + F_4(\boldsymbol{y}, \boldsymbol{y}) - 2F_4(\boldsymbol{x}, \boldsymbol{y})} &= \sqrt{\frac{2}{c} \log\cosh(\sqrt{c} \cdot d_c^{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{y}))} \\
&= \sqrt{\frac{2}{c} \log\cosh(\sqrt{c} \cdot d_{S_0}(\boldsymbol{x}, \boldsymbol{y}))}.
\end{aligned}
$$

Denote $d_{F_i} = f_i(d_{S_i})$. Since $\cosh x$ is a monotonic increasing function, f_i is monotonic increasing. And f_i is continuous, so is f_i^{-1}. Note that $d_{F_i} = 0$ when $d_{S_i} = 0$. By the property of continuous function, when $d_{F_i} \to 0$, $d_{S_i} \to 0$.

C Dataset Details

C.1 Geometric Stress Analysis

We evaluate our geometric stress analysis on the CIFAR-FS dataset, a few-shot learning benchmark derived from the CIFAR-100 dataset. CIFAR-FS consists of 100 classes, each containing 600 images of size 32×32 in RGB format. The dataset is split into 64 training classes (38,400 images), 16 validation classes (9,600 images), and 20 test classes (12,000 images), following the standard protocol (Bertinetto et al., 2019). We use this dataset for the geometric stress experiment because its moderate size and diversity allow for efficient training of multiple kernel-based prototypical networks under identical conditions.

C.2 Generalized Zero-Shot Learning (GZSL)

We evaluate on three standard GZSL benchmarks: CUB-200-2011 (Wah et al., 2011), Animals with Attributes 1 (AWA1) (Lampert et al., 2014), and Animals with Attributes 2 (AWA2) (Xian et al., 2019). Visual

features are extracted using an ImageNet pre-trained ResNet-101, yielding 2048-dimensional vectors. Semantic features are class-level attribute vectors. Table 6 summarises the dataset statistics.

Table 6: Statistics of the zero-shot learning datasets.

Dataset	Train images (seen)	Test images (seen)	Test images (unseen)	Seen classes	Unseen classes	Attribute dim.
CUB	7,057	1,764	2,967	150	50	312
AWA1	19,832	4,958	5,685	40	10	85
AWA2	23,527	5,882	7,913	40	10	85

C.3 Few-Shot Learning (FSL)

We use two widely adopted benchmarks: CUB (Wah et al., 2011) and mini-ImageNet (Deng et al., 2009). For CUB, we follow the standard split: 100 classes (5,891 images) for training, 50 classes (2,965 images) for test, and 50 classes (2,932 images) for validation. For mini-ImageNet, we use 64 classes (38,400 images) for training, 20 classes (12,000 images) for test, and 16 classes (9,600 images) for validation. All images are resized to 84×84 before being fed into the Conv-4 backbone.

C.4 Graph Node Classification

We evaluate on five benchmark attributed graphs, following the setup of Yang et al. (2023):

- **Facebook**: A mutual-like social network among verified Facebook users, collected by Rozemberczki et al. (2020) and bundled with the Karate Club library. It contains 2,489 nodes, 23,802 edges, and 4 classes, with 128-dimensional node features.
- **LastFM**: A user–artist network from the LastFM online music service, also provided by Karate Club (Rozemberczki et al., 2020). It comprises 7,624 nodes, 27,806 edges, and 18 classes (we use the same preprocessing as in prior work).
- **Terrorist**: An affiliation network of terrorists and their relations, originally introduced by Zhao et al. (2006) and available from the Network Data Repository (Rossi & Ahmed, 2015). We use the subset with 645 nodes, 2,420 edges, and 6 classes.
- **Amazon Computers (AC)**: A segment of the Amazon co-purchase graph, containing computers and related products (Shchur et al., 2018), which has 2,500 nodes, 9,554 edges, and 3 classes (the three most frequent categories).
- **Cora ML**: A citation network extracted from the Cora dataset, preprocessed by Bojchevski & Günnemann (2018). We use the version with 2,507 nodes, 5,198 edges, and 5 classes (the five largest classes).

For each dataset, we generate node embeddings using the Poincaré embedding algorithm (Nickel & Kiela, 2017) with dimensions $\dim \in \{2, 5, 10, 25\}$, following the procedure of Yang et al. (2023). Table 7 summarises the key statistics of these datasets.

C.5 Semantic Textual Similarity (STS)

We conduct experiments on the STS-Benchmark (STS-B) (Cer et al., 2017), which consists of 8,628 English sentence pairs with human-annotated similarity scores from 0 to 5. We adopt the standard split: 5,749 for training, 1,500 for validation, and 1,379 for testing. For supervised pre-training, we follow the SimCSE framework (Gao et al., 2021) and construct a mixed dataset of 275,601 triples from MNLI (Williams et al., 2018) and SNLI (Bowman et al., 2015).

Table 7: Statistics of the graph node classification datasets.

Dataset	Nodes	Edges	Classes	Feature dim.
Facebook	2,489	23,802	4	128
LastFM	7,624	27,806	18	128
Terrorist	645	2,420	6	-
Amazon Computers	2,500	9,554	3	-
Cora ML	2,507	5,198	5	-

D Experimental Details

This section provides comprehensive details of all experiments conducted in the main paper, including implementation protocols, and the loss functions used. All experiments are implemented in PyTorch and run on a single NVIDIA RTX 4090 GPU (24 GB). Random seeds are fixed per task where applicable (e.g., graph embedding seeds, data sampling seeds). The only non-standard implementation choices are the per-kernel temperature/scaling τ to align the logit range. We will release the code upon acceptance.

D.1 Projection Details

A raw encoder output $\tilde{x}$ is first clipped by

$$\tilde{x} \mapsto \min\Big(1,\ \frac{(1-\varepsilon)}{\|\tilde{x}\|}\Big) \cdot \tilde{x} \tag{20}$$

with $\varepsilon = 10^{-2}$, and then projected onto the Poincaré ball via the exponential map

$$x \mapsto \tanh\left(\sqrt{c}\|x\|\right) \cdot \frac{x}{\sqrt{c}\|x\|}$$

D.2 Temperature Details

The temperature τ scales the logits $-d_k^2(x,y)/\tau$. We tune τ on the validation set via grid search over $\{0.01, 0.05, 0.1, 0.5, 1, 2, 5, 10\}$. For each dataset, kernel, and method, we select the value achieving the best validation accuracy. All methods use the same search grid and budget. At test time, τ is fixed.

D.3 Geometric Stress Analysis

Given two sets of embeddings $\{\boldsymbol{z}_i^{(A)}\}$ and $\{\boldsymbol{z}_i^{(B)}\}$ of the same samples, stress quantifies how faithfully the pairwise distances of one embedding reproduce those of the other:

$$\text{Stress}(A \to B) = \sqrt{\frac{\sum_{i<j}\left(d_c(\boldsymbol{z}_i^{(A)}, \boldsymbol{z}_j^{(A)}) - d_c(\boldsymbol{z}_i^{(B)}, \boldsymbol{z}_j^{(B)})\right)^2}{\sum_{i<j} d_c^2(\boldsymbol{z}_i^{(A)}, \boldsymbol{z}_j^{(A)})}}. \tag{21}$$

We use the embedding trained with the hyperbolic geodesic distance as the reference (denoted "geo"). The experiment is conducted on the CIFAR-FS dataset. Following the protocol of Fang et al. (2023), we first train hyperbolic prototypical networks with a Conv-4 backbone on CIFAR-FS, using the hyperbolic geodesic distance as the similarity measure to obtain the reference embeddings. We then extract the final Poincaré embeddings of all test images. For each of our kernels and all baseline kernels, we train separate prototypical

networks that differ only in the similarity measure, extract the corresponding test embeddings, and compute the stress of each kernel-trained embedding relative to the geodesic reference. The embedding dimension is set to 64, and the curvature is fixed at $c = 0.01$ for all methods.

D.4 Generalized Zero-Shot Learning (GZSL)

We follow the kernel learning paradigm of Fang et al. (2023). Visual features $v \in \mathbb{R}^{2048}$ and semantic features $a \in \mathbb{R}^{d_a}$ are projected onto the Poincaré ball $\mathbb{D}_c^n$.

Model and pipeline. A semantic encoder (a two-layer MLP $85 \to 1200 \to 2048$ for AWA, $312 \to 1200 \to 2048$ for CUB, with BatchNorm and LeakyReLU) maps the attribute vector to a 2048-dimensional semantic embedding. The kernel is evaluated between the semantic embeddings and the visual features, scaled by τ, followed by a log-softmax and cross-entropy loss.

Optimization. We use the Adam optimizer with an initial learning rate 5×10^{-4} and weight decay 5×10^{-4}. The model is trained for 100 epochs on CUB and 30 epochs on AWA1/AWA2, with a batch size of 64. We select the curvature c achieving the best validation accuracy and c is fixed at test time. For CUB, the curvature is set to $c = 0.0001$ for RIBin, RILog, RIInt and $c = 0.01$ for RIExp. For AWA1 and AWA2, the curvature is set to $c = 0.001$ for RIBin, RILog, RIInt and $c = 0.0001$ for RIExp.

Evaluation. We report the standard ZSL accuracy (unseen classes) and the generalized ZSL (GZSL) metrics: seen accuracy S, unseen accuracy U, and their harmonic mean $\mathrm{H} = 2\mathrm{SU}/(\mathrm{S} + \mathrm{U})$.

Loss Details. We optimise a cross-entropy style loss to align visual features v with the corresponding class semantic embeddings e. For a kernel $k(\cdot, \cdot)$, the loss is given by:

$$\mathcal{L}_{\text{zsl}}^{K} = -\frac{1}{N_b} \sum_{i=1}^{N_b} \log \left(\frac{\exp(-d_k^2(e_i^*, v_i)/\tau)}{\sum_{j=1}^{|L^s|} \exp(-d_k^2(e_j, v_i)/\tau)} \right), \tag{22}$$

where N_b is the batch size, $|L^s|$ is the number of seen classes, and e_i^* is the semantic embedding of the class matching v_i. $d_k(x, y)$ is the kernel distance of k, defined by $d_k(x, y) = \sqrt{k(x, x) + k(y, y) - 2k(x, y)}$. τ is the temperature.

D.5 Few-Shot Learning (FSL)

We employ hyperbolic prototypical networks as the base framework and adopt the kernel learning paradigm of Fang et al. (2023).

Model and pipeline. We use a 4-layer convolutional backbone (Conv-4) producing a 1024-dimensional embedding, identical to the hyperbolic ProtoNet baseline. Given a support set and a query, the class prototype is the hyperbolic mean of the support embeddings; the query logits are obtained by evaluating the kernel between the query embedding and each prototype, divided by the temperature τ.

Optimization. All models are trained with Adam. On mini-ImageNet, the initial learning rate is 5×10^{-3} for Conv-4, decayed by 0.5 at epoch 80. On CUB, the initial learning rate is 10^{-3}, decayed by 0.8 at epoch 80. We train for 200 epochs, sampling 100 episodes per epoch, under 5-way 1-shot and 5-way 5-shot settings. Training episodes use 15 query samples per class and 5-way validation. We select the curvature c achieving the best validation accuracy and c is fixed at test time. For both dataset, the curvature is fixed to $c = 0.001$ for RIBin, RIExp, RIInt and $c = 0.01$ for RILog.

Evaluation. Test accuracy is averaged over 10,000 randomly sampled episodes and we report the mean $\pm$ 95% confidence interval.

Loss Details. For a C-way M-shot episode, we compute the prototype p_c for each class as the average of its M support embeddings. The loss is the negative log-likelihood of query sample q_i being assigned to its ground-truth class p^*:

$$\mathcal{L}_{\text{fsl}}^{K} = -\frac{1}{N_q}\sum_{i=1}^{N_q}\log\left(\frac{\exp(-d_k^2(q_i,p^*)/\tau)}{\sum_{c=1}^{C}\exp(-d_k^2(q_i,p_c)/\tau)}\right), \tag{23}$$

where N_q is the total number of query samples and τ is the temperature. $d_k(x,y)$ is the kernel distance of k, defined by $d_k(x,y) = \sqrt{k(x,x)+k(y,y)-2k(x,y)}$.

D.6 Graph Node Classification

We follow the experimental setup of Yang et al. (2023).

Model and pipeline. Each graph is embedded into the Poincaré ball with negative sampling; three embeddings from different random seeds are averaged. A kernel Gram matrix is then computed from the embeddings and fed to a kernel SVM. We evaluate the four kernels (RILog, RIBin, RIInt, RIExp) with embedding dimensions $\{2, 5, 10, 25\}$.

Optimization. The embedding is learned by burn-in Riemannian SGD (150 epochs, learning rate 0.5) followed by Riemannian Adam with cosine decay (800–1200 epochs, learning rate 0.2). The SVM regularization $C \in \{0.1, 1, 10, 100\}$ and the kernel curvature $c \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.8, 1.0, 1.5, 2.0, 3.0, 5.0, 10.0, 20.0, 50.0, 100.0\}$ are selected by inner 3-fold cross-validation on a 500-node subset.

Evaluation. The final accuracy is obtained by 5-fold stratified cross-validation, reported as mean $\pm$ standard deviation.

Loss Details. For graph node classification, we use the standard SVM loss in the reproducing kernel Hilbert space (RKHS). The primal problem for a kernel k is:

$$\min_{w}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\max(0, 1-y_i f(z_i)), \tag{24}$$

with $f(z) = w^T\phi(z) + b$ and $k(z_i, z_j) = \langle\phi(z_i), \phi(z_j)\rangle$. The dual formulation is solved using LIBSVM.

D.7 Semantic Textual Similarity (STS)

Models and pipeline. We follow the SimCSE (Gao et al., 2021) framework for supervised contrastive pre-training, using BERT-base-uncased (Devlin et al., 2019) as the backbone. The CLS token embedding from BERT is projected onto the Poincaré ball. The model is trained on the NLI dataset using the supervised contrastive objective, with the cosine similarity replaced by the proposed kernels. The curvature is fixed at $c = 0.1$.

Optimization. We use AdamW with learning rate 3×10^{-5}, dropout 0.1, batch size 64, and 2 training epochs.

Evaluation. We compute the Spearman correlation between the predicted similarity scores and human annotations. The model checkpoint achieving the highest Spearman correlation on the validation set is selected. We then report its Spearman correlation on the STS-B test set

Loss Details. We adopt a supervised contrastive loss. For each anchor h_i in a batch, its positive is h_i^+ and negatives are other samples h_j^- in the batch. The loss is:

$$\mathcal{L}_{\text{sts}}^K = -\log \frac{\exp\left(-d_k^2(h_i, h_i^+)/\tau\right)}{\sum_{j=1}^B \exp\left(-d_k^2(h_i, h_j^{\{+,-\}})/\tau\right)}, \tag{25}$$

where B is the batch size and τ is the temperature. $d_k(x, y)$ is the kernel distance of k, defined by $d_k(x, y) = \sqrt{k(x, x) + k(y, y) - 2k(x, y)}$.

E Robustness of kernel machines

We evaluate the robustness of the proposed kernels on GZSL for datasets CUB. We compare our kernels with different curvature of hyperbolic geometry ($c = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10$). When comparing the robustness with c, λ is fixed to 0.001 for RIExp and t is also fixed to 0.001 for RIBin. We also evaluate the sensitivity of t with c fixed to 0.0001 and λ with $c = 0.01$.

Fig. 1 shows the sensitivity of the harmonic mean H to the curvature parameter c. Over the range $c \in [10^{-5}, 0.1]$, H varies in a small range, showing the four kernels are robust against the curvature. However, the performance degrades when c is too large ($c > 0.1$). This indicates that four kernels are sensitive to large c.

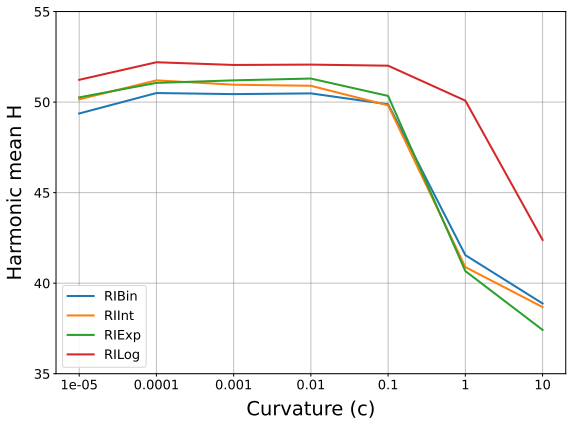
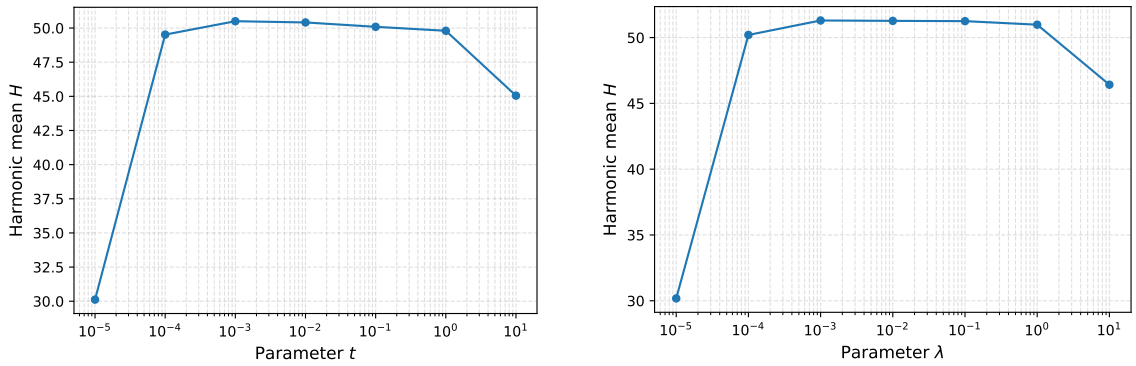


Figure 1: The performance on CUB with various curvature of the hyperbolic geometry. The experiment is conducted on a GZSL task. Best view in color.

Fig. 2 shows the sensitivity of the harmonic mean H to the parameters t and λ. For both parameters, H is approximately 30 at 10^{-5}, increases sharply to around 50 at 10^{-4}, remains stable near 51 for values from 10^{-3} to 1, and then decreases to about 46 at 10. The left panel corresponds to t, and the right panel corresponds to λ. These results indicate that the model is robust to both t and λ over the range $[10^{-3}, 1]$, whereas extremely small or large values degrade performance. We therefore set $t = 10^{-3}$ and $\lambda = 10^{-3}$ in all experiments.



(1) t is set to $10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10$. (2) λ is set to $10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10$.

Figure 2: The performance with parameter t for RIBin and λ for RIExp.

F Numerical Stability Analysis of the Riemannian Isometric Logarithmic Kernel

This appendix studies the optimization properties of RILog in hyperbolic prototypical networks. We first review the advantages and disadvantages of clipping, then give the formal definition of few-shot learning and hyperbolic prototypical networks, derive the gradient of the kernelized loss, and finally establish the boundary stability of RILog.

F.1 Analysis of the Advantages and Disadvantages of Clipping

The clipping map is defined by Equation Equation 20 and is designed to alleviate the vanishing-gradient problem in hyperbolic geometry. Given a hyperbolic embedding $\boldsymbol{x}^H$ in a unit Poincaré ball with $c = 1$ and a loss function L, the gradient with respect to the Euclidean embedding contains the factor

$$\frac{\partial L}{\partial \boldsymbol{x}^H} = \frac{(1 - \|\boldsymbol{x}\|^2)^2}{4} \nabla L(\boldsymbol{x}^H),$$

where $\nabla L(\boldsymbol{x}^H)$ is the Euclidean gradient. During training, the points move toward the boundary of the Poincaré ball, resulting in $\|\boldsymbol{x}^H\| \to 1$. Therefore, the gradient factor $(1 - \|\boldsymbol{x}\|^2)^2$ diminishes during training, which indicates vanishing gradients. One approach is to apply soft regularization to the Euclidean embedding before the exponential map, constraining its norm to stay under a certain threshold. This leads to the clipping map:

$$\mathrm{CLIP}(\tilde{\boldsymbol{x}}, \varepsilon) = \min\left(1, \frac{1 - \varepsilon}{\|\tilde{\boldsymbol{x}}\|}\right) \tilde{\boldsymbol{x}}.$$

An obvious advantage of clipping is that it can avoid the problem of vanishing gradients. However, hyperbolic embeddings after being clipped will not lie throughout the hyperbolic space, indicating that the embeddings do not use whole geometry. Since the volume of hyperbolic space grows exponentially with radius, the space with most of the volume is actually near the boundary. Therefore, the clipping can cause a certain degree of data distortion. If one method can avoid gradient explosion or vanishing without using the clipping, it will gain better performance than those methods using the clipping. Training with RILog is indeed such a method that we will prove in the following subsections.

F.2 Formal Definition of Few-Shot Learning and Hyperbolic Prototypical Networks

We only prove our conclusions for the few-shot learning task. Here we give its formal definition for the convenience of our derivation.

Definition F.1 (N-way K-shot classification problem)**.** *Let $\mathcal{D}_{\text{base}}$ be the base-class dataset and $\mathcal{D}_{\text{novel}}$ the novel-class dataset, and assume that their label sets are disjoint, i.e.,*

$$\mathcal{Y}_{\text{base}} \cap \mathcal{Y}_{\text{novel}} = \varnothing.$$

The N-way K-shot classification task is defined as follows:

- *Support set $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N\times K}$: for each of the N novel classes, randomly sample K labeled examples.*
- *Query set $\mathcal{Q} = \{(\tilde{\boldsymbol{x}}_j, \tilde{y}_j)\}_{j=1}^{N\times Q}$: from the same N novel classes, randomly sample Q examples for evaluation.*

The model must predict the class labels of the query samples $\tilde{\boldsymbol{x}}_j$ using only the information in the support set $\mathcal{S}$.

Hyperbolic prototypical networks are the hyperbolic extension of prototypical networks. For each class $l \in \{1, \dots, N\}$, its prototype is defined as the hyperbolic mean of the support samples of that class in the Poincaré ball:

$$\boldsymbol{p}_l = \mathrm{HypAve}\left(\{\exp_0(f_\omega(\boldsymbol{x}_i)) : (\boldsymbol{x}_i, y_i) \in \mathcal{S}_l\}\right),$$

where f_ω is the embedding network with parameters ω, $\exp_0$ is the exponential map at the origin, and HypAve is the hyperbolic mean defined by

$$\mathrm{HypAve}(\{\boldsymbol{z}_i\}_{i=1}^{N_l}) = \frac{\sum_{i=1}^{N_l} \gamma_i \boldsymbol{z}_i}{\sum_{i=1}^{N_l} \gamma_i}, \qquad \boldsymbol{z}_i \in \mathcal{D}_c^n,$$

where

$$\gamma_i = \frac{1}{\sqrt{1 - c\|\boldsymbol{z}_i\|^2}}$$

is the Lorentz factor, and $N_l = |\mathcal{S}_l|$ is the number of support samples of class l.

The posterior probability that a query sample $\boldsymbol{x}$ belongs to class l is given by the softmax over distances in the embedding space:

$$p_\omega(y = l \mid \boldsymbol{x}) = \frac{\exp(-d_c(\exp_0(f_\omega(\boldsymbol{x})), \boldsymbol{p}_l))}{\sum_{m=1}^{N} \exp(-d_c(\exp_0(f_\omega(\boldsymbol{x})), \boldsymbol{p}_m))},$$

where $d_c(\cdot, \cdot)$ is the distance on the Poincaré ball of curvature $-c$.

F.3 Gradient of the Kernelized Loss Function

Let the feature extractor be $f_\omega : \mathcal{X} \to \mathbb{R}^n$ with parameters ω, and let the Poincaré-ball embedding of a sample $\boldsymbol{x}$ be

$$\boldsymbol{z} = \exp_0(f_\omega(\boldsymbol{x})).$$

For an N-way K-shot task, denote the prototype of class l by $\boldsymbol{p}_l$. We use a kernel

$$k : \mathcal{D}_c^n \times \mathcal{D}_c^n \to \mathbb{R}$$

as the similarity measure. For a kernel k, the kernel-induced distance is defined by

$$d_k(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{k(\boldsymbol{x}, \boldsymbol{x}) + k(\boldsymbol{y}, \boldsymbol{y}) - 2k(\boldsymbol{x}, \boldsymbol{y})}.$$

Then the predicted probability that a query sample $\boldsymbol{x}_q$ belongs to class l is

$$P_q(l) = \frac{\exp(-d_k^2(\boldsymbol{z}_q, \boldsymbol{p}_l))}{\sum_{m=1}^{N} \exp(-d_k^2(\boldsymbol{z}_q, \boldsymbol{p}_m))}, \qquad \boldsymbol{z}_q = \exp_0(f_\omega(\boldsymbol{x}_q)).$$

The loss function is the average cross-entropy over the query set:

$$\mathcal{L}(\omega) = -\frac{1}{|\mathcal{Q}|} \sum_{(\boldsymbol{x}_q, y_q) \in \mathcal{Q}} \log P_q(y_q).$$

During training, the prototypes $\boldsymbol{p}_l$ are usually treated as constants in order to reduce computational complexity (stop-gradient operation). The gradient of the loss with respect to ω is backpropagated only through the query embeddings $\boldsymbol{z}_q$. By the chain rule,

$$\nabla_\omega \mathcal{L} = \frac{1}{|\mathcal{Q}|} \sum_{\boldsymbol{z}_q \in \mathcal{Q}} \sum_{l=1}^{N} \big(P_q(l) - \delta_{l,y_q}\big) \frac{\partial(-d_k^2)}{\partial \boldsymbol{z}}(\boldsymbol{z}_q, \boldsymbol{p}_l) J(\boldsymbol{z}_q) \frac{\partial f_\omega(\boldsymbol{z}_q)}{\partial \omega}.$$

Here $\partial(-d_k^2)/\partial \boldsymbol{z}$ is the row gradient of the negative squared kernel distance with respect to its first argument, δ_{l,y_q} is the Kronecker delta, and $J(\boldsymbol{z})$ is the Jacobian matrix of the exponential map $\partial \exp_0(\boldsymbol{v})/\partial \boldsymbol{v}$ evaluated at $\boldsymbol{z} = \exp_0(\boldsymbol{v})$. It can be written directly as a function on the Poincaré ball:

$$J(\boldsymbol{z}) = (1 - c\|\boldsymbol{z}\|^2) \frac{\boldsymbol{z}\boldsymbol{z}^\top}{\|\boldsymbol{z}\|^2} + \frac{\sqrt{c}\|\boldsymbol{z}\|}{\tanh^{-1}(\sqrt{c}\|\boldsymbol{z}\|)} \left(I_n - \frac{\boldsymbol{z}\boldsymbol{z}^\top}{\|\boldsymbol{z}\|^2} \right),$$

with the convention $J(0) = I_n$. This expression decomposes the Jacobian of the exponential map into radial and tangential parts, where the radial scaling factor is $1 - c\|\boldsymbol{z}\|^2$, and the tangential scaling factor is

$$\frac{\sqrt{c}\|\boldsymbol{z}\|}{\tanh^{-1}(\sqrt{c}\|\boldsymbol{z}\|)}.$$

F.4 Training Stability Analysis

In the gradient expression

$$\nabla_\omega \mathcal{L} = \frac{1}{|\mathcal{Q}|} \sum_{\boldsymbol{z}_q \in \mathcal{Q}} \sum_{l=1}^{N} \big(P_q(l) - \delta_{l,y_q}\big) \psi_{\mathcal{B}}(\boldsymbol{z}_q, \boldsymbol{p}_l) \frac{\partial f_\omega(\boldsymbol{z}_q)}{\partial \omega},$$

the error signal

$$P_q(l) - \delta_{l,y_q} \in [-2, 2]$$

is always bounded, and the Jacobian matrix of the backbone network $\partial f_\omega / \partial \omega$ is usually numerically stable in commonly used architectures. Therefore, numerical stability is essentially determined by the kernel composite gradient:

$$\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p}) = \frac{\partial(-d_k^2)}{\partial \boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{p}) J(\boldsymbol{z}), \qquad \boldsymbol{z} \in \mathcal{D}_c^n.$$

This term couples the gradient of the kernel on the manifold with the Jacobian of the exponential map, and describes the gradient after composing the kernel with the exponential map.

The expression for $J(\boldsymbol{z})$ reveals the source of training instability near the boundary of the Poincaré ball: as $\|\boldsymbol{z}\| \to 1/\sqrt{c}$, the radial coefficient

$$1 - c\|\boldsymbol{z}\|^2 \to 0,$$

and the tangential coefficient

$$\frac{\sqrt{c}\|\boldsymbol{z}\|}{\tanh^{-1}(\sqrt{c}\|\boldsymbol{z}\|)} \to 0.$$

If the growth rate of the kernel gradient $\partial k/\partial \boldsymbol{u}$ in different directions cannot precisely compensate for these decays, then the composite gradient $\psi_{\mathcal{B}}$ will suffer from gradient vanishing. We therefore restrict our stability requirement to prototypes lying in a strictly interior ball, which is the natural regime for prototypes formed as hyperbolic means of finitely many support embeddings.

Definition F.2 (Stable training)**.** *A kernel k makes training stable in the hyperbolic prototypical network framework if the following conditions hold:*

1. ***Boundedness on strictly interior prototypes.*** *For every $R < 1/\sqrt{c}$, there exists a constant $M(R) < \infty$ such that for all $\boldsymbol{z} \in \mathcal{D}_c^n$ and all $\boldsymbol{p} \in \mathcal{D}_c^n$ with $\|\boldsymbol{p}\| \le R$,*

$$\|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\| \le M(R).$$

2. ***Boundary non-vanishing.*** *For every fixed $\boldsymbol{p} \in \mathcal{D}_c^n$,*

$$\lim_{\|\boldsymbol{z}\| \to 1/\sqrt{c}} \|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\| > 0.$$

3. ***Positive lower bound away from the prototype.*** *For every $R < 1/\sqrt{c}$ and every $\varepsilon > 0$, there exists $m(R, \varepsilon) > 0$ such that whenever $\|\boldsymbol{p}\| \le R$ and $\|\boldsymbol{z} - \boldsymbol{p}\| \ge \varepsilon$,*

$$\|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\| \ge m(R, \varepsilon).$$

The upper bound in condition (1) prevents gradient explosion whenever the prototypes stay in a strictly interior ball. Condition (2) prevents gradient vanishing when the embedding point is driven by the optimizer toward the ball boundary. Condition (3) ensures that the gradient is not abnormally small away from the prototype. We deliberately do not require a global positive lower bound, because as $\boldsymbol{z} \to \boldsymbol{p}$ the squared kernel distance attains its minimum and its gradient necessarily tends to zero; hence a global bound $m > 0$ is impossible. We also do not require a global upper bound over all $\boldsymbol{p} \in \mathcal{D}_c^n$: when both $\boldsymbol{z}$ and $\boldsymbol{p}$ approach the boundary simultaneously with a small tangential offset, the tangential component of $\psi_{\mathcal{B}}$ can blow up, as we discuss after Theorem F.1.

F.5 Stability Analysis of the Riemannian Isometric Logarithmic Kernel

Recall that RILog is defined by

$$k^{\log}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{c}\left[-k^{\text{base}}(\boldsymbol{x}, \boldsymbol{y}) + k^{\text{base}}(\boldsymbol{x}, \boldsymbol{x}_0) + k^{\text{base}}(\boldsymbol{x}_0, \boldsymbol{y})\right],$$

where

$$k^{\text{base}}(\boldsymbol{x}, \boldsymbol{y}) = \log(1 - c\rho(\boldsymbol{x}, \boldsymbol{y})), \qquad \rho(\boldsymbol{x}, \boldsymbol{y}) = -\frac{2\|\boldsymbol{x} - \boldsymbol{y}\|^2}{(1 - c\|\boldsymbol{x}\|^2)(1 - c\|\boldsymbol{y}\|^2)}.$$

For RILog, the squared kernel-induced distance is

$$d_k^2(\boldsymbol{x}, \boldsymbol{y}) = k^{\log}(\boldsymbol{x}, \boldsymbol{x}) + k^{\log}(\boldsymbol{y}, \boldsymbol{y}) - 2k^{\log}(\boldsymbol{x}, \boldsymbol{y}) = \frac{2}{c}\log(1 - c\rho(\boldsymbol{x}, \boldsymbol{y})).$$

Note that $\log(1 - c\rho(\boldsymbol{x}, \boldsymbol{y})) \geq 0$ and

$$1 - c\rho(\boldsymbol{x}, \boldsymbol{y}) = \cosh(\sqrt{c}\, d_c^D(\boldsymbol{x}, \boldsymbol{y})).$$

We now compute the composite gradient $\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})$. Let $\boldsymbol{z}, \boldsymbol{p} \in \mathcal{D}_c^n$ and set $r = \|\boldsymbol{z}\|$, $\hat{\boldsymbol{z}} = \boldsymbol{z}/r$. A direct calculation gives

$$\frac{\partial(-d_k^2)}{\partial \boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{p}) = -\frac{8}{(1 - c\|\boldsymbol{z}\|^2)(1 - c\|\boldsymbol{p}\|^2) + 2c\|\boldsymbol{z} - \boldsymbol{p}\|^2} \left[(\boldsymbol{z} - \boldsymbol{p}) + \frac{c\|\boldsymbol{z} - \boldsymbol{p}\|^2}{1 - c\|\boldsymbol{z}\|^2}\boldsymbol{z}\right]^\top.$$

The Jacobian of the exponential map is

$$J(\boldsymbol{z}) = (1 - cr^2)\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top + \frac{\sqrt{c}r}{\tanh^{-1}(\sqrt{c}r)}\left(I_n - \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top\right).$$

Therefore, the composite gradient can be decomposed into radial and tangential parts:

$$\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p}) = \psi_{\mathcal{B}}^{\parallel}(\boldsymbol{z}, \boldsymbol{p}) + \psi_{\mathcal{B}}^{\perp}(\boldsymbol{z}, \boldsymbol{p}),$$

where

$$\psi_{\mathcal{B}}^{\parallel}(\boldsymbol{z}, \boldsymbol{p}) = (1 - cr^2)\left(\frac{\partial(-d_k^2)}{\partial \boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{p}) \cdot \hat{\boldsymbol{z}}\right)\hat{\boldsymbol{z}}^\top,$$

and

$$\psi_{\mathcal{B}}^{\perp}(\boldsymbol{z}, \boldsymbol{p}) = \frac{\sqrt{c}r}{\tanh^{-1}(\sqrt{c}r)}\frac{\partial(-d_k^2)}{\partial \boldsymbol{z}}(\boldsymbol{z}, \boldsymbol{p})\left(I_n - \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top\right).$$

Simplifying the radial part using $\boldsymbol{z} \cdot \hat{\boldsymbol{z}} = r$ and $(\boldsymbol{z} - \boldsymbol{p}) \cdot \hat{\boldsymbol{z}} = r - \boldsymbol{p} \cdot \hat{\boldsymbol{z}}$, we obtain

$$\psi_{\mathcal{B}}^{\parallel} = -\frac{8}{(1 - cr^2)(1 - c\|\boldsymbol{p}\|^2) + 2c\|\boldsymbol{z} - \boldsymbol{p}\|^2}\left[(1 - cr^2)(r - \boldsymbol{p} \cdot \hat{\boldsymbol{z}}) + cr\|\boldsymbol{z} - \boldsymbol{p}\|^2\right]\hat{\boldsymbol{z}}^\top.$$

For the tangential part, since $\boldsymbol{z}^\top(I_n - \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top) = 0$, we get

$$\psi_{\mathcal{B}}^{\perp} = \frac{\sqrt{c}r}{\tanh^{-1}(\sqrt{c}r)}\frac{8}{(1 - cr^2)(1 - c\|\boldsymbol{p}\|^2) + 2c\|\boldsymbol{z} - \boldsymbol{p}\|^2}\boldsymbol{p}^\top\left(I_n - \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top\right).$$

Theorem F.1 (Boundary stability of RILog)**.** *Let $k^{\log}$ be RILog and fix $R < 1/\sqrt{c}$. Then the composite gradient $\psi_{\mathcal{B}}$ satisfies the following properties:*

1. ***Boundedness on strictly interior prototypes.*** *There exists a constant $M(R) < \infty$ such that for all $\boldsymbol{z} \in \mathcal{D}_c^n$ and all $\boldsymbol{p} \in \mathcal{D}_c^n$ with $\|\boldsymbol{p}\| \leq R$,*
$$\|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\| \leq M(R).$$
2. ***Boundary non-vanishing.*** *For every fixed $\boldsymbol{p} \in \mathcal{D}_c^n$ with $\|\boldsymbol{p}\| \leq R$,*
$$\lim_{\|\boldsymbol{z}\| \to 1/\sqrt{c}} \|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\| = \frac{4}{\sqrt{c}} > 0.$$
3. ***Positive lower bound away from the prototype.*** *For every $\varepsilon > 0$, there exists $m(R, \varepsilon) > 0$ such that whenever $\|\boldsymbol{p}\| \leq R$ and $\|\boldsymbol{z} - \boldsymbol{p}\| \geq \varepsilon$,*
$$\|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\| \geq m(R, \varepsilon).$$

Proof. We prove the three statements separately.

(1) Boundedness on strictly interior prototypes. Fix $R < 1/\sqrt{c}$ and consider the set

$$K_R = \{(\boldsymbol{z}, \boldsymbol{p}) : \boldsymbol{z} \in \mathcal{D}_c^n,\ \|\boldsymbol{p}\| \le R\}.$$

The composite gradient $\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})$ is continuous on the interior of K_R: the kernel distance d_k^2 is smooth on $\mathcal{D}_c^n \times \mathcal{D}_c^n$, and $J(\boldsymbol{z})$ is continuous on $\mathcal{D}_c^n$. The only possible source of discontinuity is the boundary $\|\boldsymbol{z}\| \to 1/\sqrt{c}$. We now show that $\psi_{\mathcal{B}}$ extends continuously to the closed set

$$\overline{K}_R = \left\{(\boldsymbol{z}, \boldsymbol{p}) : \|\boldsymbol{z}\| \le 1/\sqrt{c},\ \|\boldsymbol{p}\| \le R\right\}.$$

Fix $\boldsymbol{p}$ with $\|\boldsymbol{p}\| \le R$. As $r = \|\boldsymbol{z}\| \to 1/\sqrt{c}$, we have $1 - cr^2 \to 0$ and, since $\boldsymbol{p}$ is strictly interior,

$$\left\| \frac{1}{\sqrt{c}}\hat{\boldsymbol{z}} - \boldsymbol{p} \right\| \ge \frac{1}{\sqrt{c}} - R > 0.$$

For the radial component, the numerator term $(1 - cr^2)(r - \boldsymbol{p} \cdot \hat{\boldsymbol{z}})$ tends to zero, while the term $cr\|\boldsymbol{z} - \boldsymbol{p}\|^2$ tends to

$$\sqrt{c}\left\| \frac{1}{\sqrt{c}}\hat{\boldsymbol{z}} - \boldsymbol{p} \right\|^2,$$

and the denominator tends to

$$2c\left\| \frac{1}{\sqrt{c}}\hat{\boldsymbol{z}} - \boldsymbol{p} \right\|^2 > 0.$$

Hence

$$\psi_{\mathcal{B}}^{\parallel} \to -\frac{4}{\sqrt{c}}\hat{\boldsymbol{z}}^\top.$$

For the tangential component, the fraction preceding $\boldsymbol{p}^\top(I_n - \hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top)$ tends to a finite value, while

$$\frac{\sqrt{c}r}{\tanh^{-1}(\sqrt{c}r)} \to 0,$$

so

$$\psi_{\mathcal{B}}^{\perp} \to 0.$$

Therefore, along any fixed radial direction $\hat{\boldsymbol{z}}$, the boundary limit exists and equals $-(4/\sqrt{c})\hat{\boldsymbol{z}}^\top$, whose norm is $4/\sqrt{c}$. Defining $\psi_{\mathcal{B}}$ on $\{\|\boldsymbol{z}\| = 1/\sqrt{c}\}$ by this limit yields a continuous extension on the compact set $\overline{K}_R$, which therefore attains a finite maximum $M(R)$.

(2) Boundary non-vanishing. This follows directly from the computation in part (1): for every fixed $\boldsymbol{p}$ with $\|\boldsymbol{p}\| \le R$,

$$\lim_{\|\boldsymbol{z}\| \to 1/\sqrt{c}} \|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\| = \frac{4}{\sqrt{c}} > 0.$$

(3) Positive lower bound away from the prototype. Fix $R < 1/\sqrt{c}$ and $\varepsilon > 0$. Consider the compact set

$$K_{R,\varepsilon} = \left\{(\boldsymbol{z}, \boldsymbol{p}) : \|\boldsymbol{z}\| \le 1/\sqrt{c},\ \|\boldsymbol{p}\| \le R,\ \|\boldsymbol{z} - \boldsymbol{p}\| \ge \varepsilon\right\}.$$

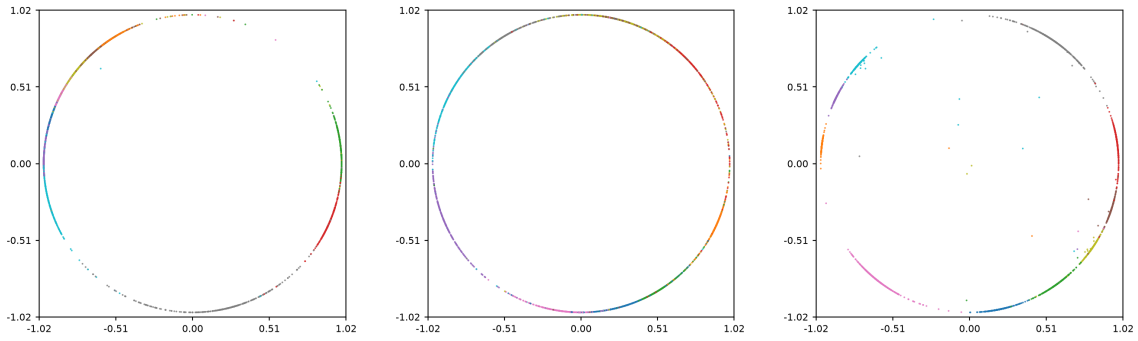
The trivial nonnegativity of the norm only gives $\Phi \ge 0$, which is not enough to prevent gradient vanishing. What we need is a strictly positive lower bound. On $K_{R,\varepsilon}$, the composite gradient $\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})$ never vanishes. Indeed, for $\|\boldsymbol{z}\| < 1/\sqrt{c}$, the Jacobian $J(\boldsymbol{z})$ is invertible, and $\partial(-d_k^2)/\partial \boldsymbol{z}$ vanishes only at $\boldsymbol{z} = \boldsymbol{p}$, which is excluded by $\|\boldsymbol{z} - \boldsymbol{p}\| \ge \varepsilon$. On the boundary $\|\boldsymbol{z}\| = 1/\sqrt{c}$, part (2) shows that $\psi_{\mathcal{B}}$ is non-zero. Hence $\Phi(\boldsymbol{z}, \boldsymbol{p}) = \|\psi_{\mathcal{B}}(\boldsymbol{z}, \boldsymbol{p})\|$ is strictly positive on the compact set $K_{R,\varepsilon}$. Since Φ is continuous there, it attains a positive minimum $m(R, \varepsilon) > 0$. □

Remark. The boundedness conclusion of Theorem F.1 genuinely requires $\|\boldsymbol{p}\| \leq R < 1/\sqrt{c}$. If $\boldsymbol{p}$ is allowed to approach the boundary simultaneously with $\boldsymbol{z}$, the tangential component can diverge. Thus no global finite upper bound M over all $\boldsymbol{z}, \boldsymbol{p} \in \mathcal{D}_c^n$ can exist, and any stability claim must restrict the prototypes to a strictly interior ball. In practice this is a mild restriction: prototypes are hyperbolic means of finitely many support embeddings, and the support embeddings are produced by a finite network evaluated on finite data, so they remain in a compact subset of $\mathcal{D}_c^n$ throughout training.

Theorem F.1 reveals the compensation mechanism of RILog on the Poincaré ball. The factor $(1 - c\|\boldsymbol{z}\|^2)$ in the denominator of the kernel gradient is exactly canceled by the same factor in the radial part of $J(\boldsymbol{z})$, so that the radial composite gradient tends to a non-zero constant near the boundary, thereby effectively avoiding gradient vanishing and gradient explosion. Although the theoretical analysis suggests stability without explicit clipping, we did not conduct a systematic no-clipping ablation. Therefore, this stability claim is supported only theoretically and is not empirically verified here. Future work will report clipped-versus-unclipped comparisons, including gradient norms, divergence rates, and performance differences.

G VISUALIZATION OF KERNEL EMBEDDING FEATURES

In this section, we visualize the features extracted by no-kernel method, AHL, and RILog on MNIST dataset (LeCun et al., 1998). Following the pipeline in Fang et al. (2023) for kernel method, we first generate the feature vector in hyperbolic space, and then map the vector to the RKHS by the proposed kernels. The no-kernel method uses geodesic distance to classify the features. Both 2-D embeddings are trained by CNN consisting of four convolutional layers and two fully connected layers. The curvature is fixed to $c = 0.1$.



(a) no-kernel (b) AHL (c) RILog

Figure 3: Visualization of the features in hyperbolic spaces learned for MNIST. Best viewed in color

From Fig. 3, we can find in both pipelines, most samples are distributed near the boundary. This is reasonable due to the geometric property of Poincaré ball that the volume of the ball grows exponentially with its radius. Therefore, the features clustered at the boundary indicate full utilization of the volume. In Fig. 4, we further visualize the feature embeddings for each class. We observe that most embeddings from our kernel are more compact than the other two, for example, Fig.(4), Fig.(5) vs. Fig.(6) and Fig.(28), Fig.(29) vs. Fig.(30). Furthermore, the other methods exhibit significant overlap between different classes, whereas our kernel shows almost no overlap. The above two facts demonstrate that our kernel exhibits superior classification performance.

H DETAILED TABLES

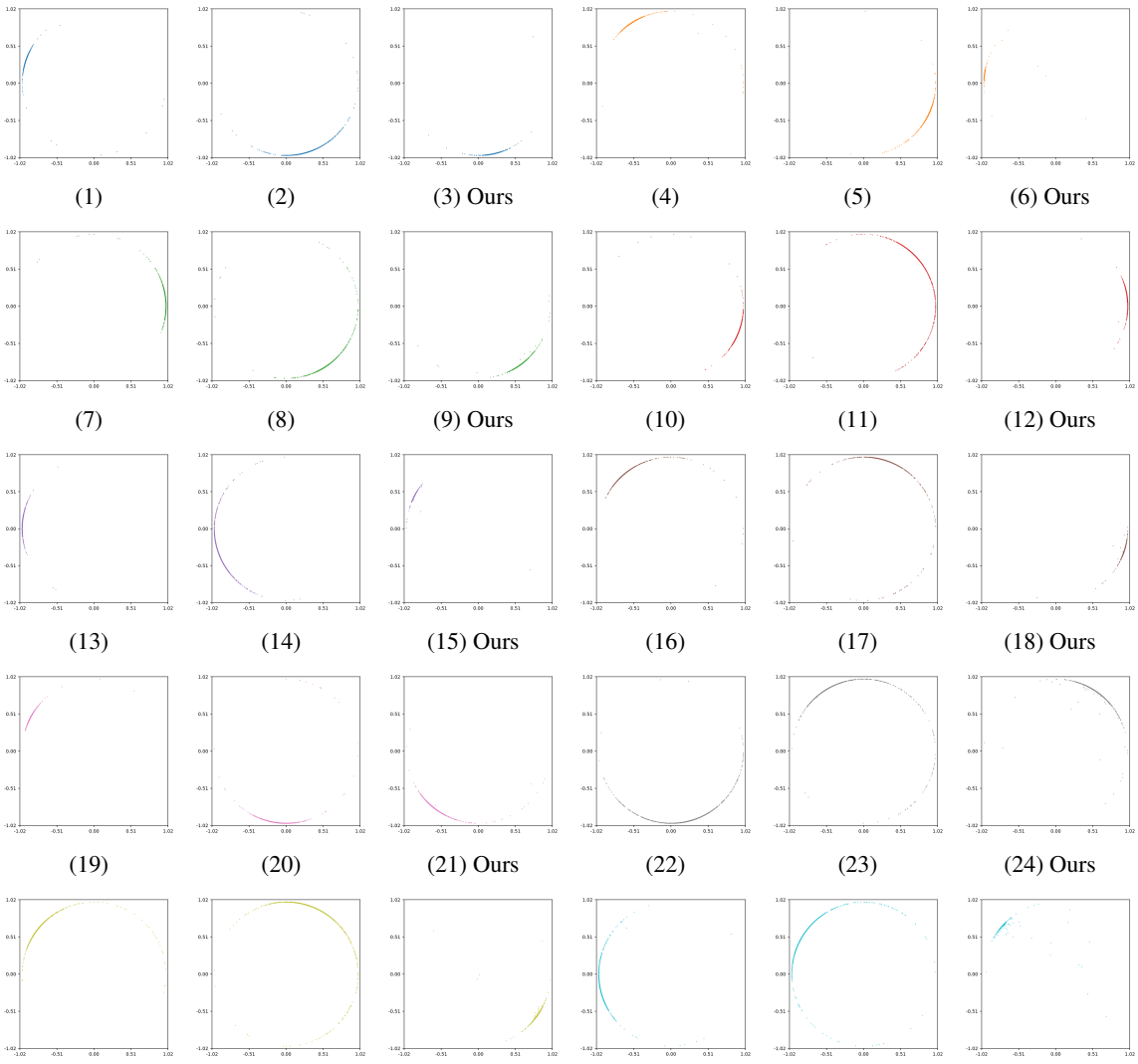


Figure 4: Visualization of the feature embeddings for each class on MNIST dataset. Best viewed in color.

Table 8: Geometric stress of embeddings trained by different similarity measures, relative to the geodesic-distance reference.

	Cosine	PTang	PRBF	PLap	PBin	PRad	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
Stress(geo$\to k$)	0.2995	0.2341	0.3120	0.0412	0.6726	0.6347	0.6979	0.6999	0.6961	0.6834	0.5481	0.5506	0.1623	0.1656	0.2792	0.0399	0.0398	0.0404	**0.0392**

Table 9: GZSL results (%) on CUB, AWA1 and AWA2. Best H per dataset is in **bold**.

Dataset	Metric	Baseline	PTang	PRBF	PLap	PBin	PRad	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
AWA1	U	29.8	52.3	59.0	60.7	43.7	60.2	51.2	51.3	56.3	53.4	55.4	58.8	59.3	56.8	64.9	59.0	56.9	59.0	71.6
	S	76.4	85.2	84.6	83.5	88.9	86.7	84.7	85.4	82.7	88.9	85.3	85.3	85.2	82.5	83.7	83.4	84.4	84.4	76.6
	H	42.9	64.8	69.5	70.3	58.6	71.1	63.8	64.1	67.0	66.7	67.2	69.6	69.9	67.3	73.1	69.1	68.0	69.4	**74.0**
AWA2	U	25.5	37.1	42.9	54.1	39.8	48.2	44.5	42.2	45.1	44.9	49.4	46.2	48.3	47.1	69.3	50.4	51.0	50.2	73.9
	S	76.4	88.5	89.5	87.1	90.5	90.3	90.8	90.9	90.1	90.9	89.4	87.6	88.6	86.5	84.0	86.6	87.9	87.8	78.9
	H	38.2	52.3	57.9	66.7	55.4	62.8	59.8	57.6	60.1	60.1	63.6	60.5	62.5	61.0	75.9	63.7	64.6	63.9	**76.3**
CUB	U	18.6	40.8	44.6	46.2	39.8	45.8	43.3	41.7	45.0	45.2	46.2	49.0	44.7	47.4	49.0	51.0	49.9	49.5	49.1
	S	44.6	58.1	57.8	56.1	56.9	57.6	58.3	58.9	56.7	56.1	56.1	52.8	58.3	55.3	54.0	50.1	52.9	53.1	55.7
	H	26.3	47.9	50.3	50.7	46.8	51.0	49.7	48.8	50.1	50.1	50.7	50.8	50.6	51.0	51.4	50.5	51.3	51.2	**52.2**

Table 10: Few-shot classification accuracy (%) with 95% confidence interval on CUB and mini-ImageNet datasets. Best result for each dataset and experimental setting is highlighted in **bold**.

Dataset	Setting	Baseline	PTang	PRBF	PLap	PBin	PRad	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
CUB	1-shot	$59.6_{0.2}$	$60.1_{0.3}$	$61.4_{0.2}$	$62.9_{0.2}$	$62.6_{0.2}$	$65.6_{0.2}$	$65.8_{0.2}$	$62.3_{0.2}$	$56.7_{0.2}$	$56.9_{0.2}$	$61.2_{0.2}$	$62.0_{0.2}$	$60.6_{0.2}$	$60.9_{0.2}$	$63.9_{0.2}$	$67.3_{0.2}$	$67.3_{0.2}$	$65.8_{0.2}$	$\mathbf{68.6}_{0.2}$
	5-shot	$78.3_{0.2}$	$82.0_{0.2}$	$82.7_{0.2}$	$81.7_{0.2}$	$83.0_{0.1}$	$82.4_{0.2}$	$82.9_{0.2}$	$80.3_{0.2}$	$79.9_{0.2}$	$79.1_{0.2}$	$82.7_{0.1}$	$82.2_{0.2}$	$82.2_{0.2}$	$81.6_{0.2}$	$83.3_{0.2}$	$\mathbf{85.1}_{0.2}$	$83.8_{0.2}$	$83.6_{0.2}$	$\mathbf{85.1}_{0.2}$
mini-ImageNet	1-shot	$52.7_{0.2}$	$54.0_{0.2}$	$54.4_{0.2}$	$53.1_{0.2}$	$53.4_{0.2}$	$53.6_{0.2}$	$53.6_{0.2}$	$52.6_{0.2}$	$53.5_{0.2}$	$52.7_{0.2}$	$52.5_{0.2}$	$53.8_{0.2}$	$52.4_{0.2}$	$53.0_{0.2}$	$54.6_{0.2}$	$55.2_{0.2}$	$54.3_{0.2}$	$54.5_{0.2}$	$\mathbf{56.3}_{0.2}$
	5-shot	$71.7_{0.2}$	$73.1_{0.2}$	$73.1_{0.2}$	$71.3_{0.2}$	$72.6_{0.2}$	$72.9_{0.2}$	$72.6_{0.2}$	$71.3_{0.2}$	$71.9_{0.2}$	$71.5_{0.2}$	$71.7_{0.2}$	$71.7_{0.2}$	$72.0_{0.2}$	$71.5_{0.2}$	$73.2_{0.2}$	$72.8_{0.2}$	$71.9_{0.2}$	$71.8_{0.2}$	$\mathbf{73.8}_{0.2}$

Table 11: Mean accuracy (%) of node classification on graph datasets. The subscript of each number indicates the corresponding standard deviation. Best results are highlighted in **bold**

Dataset	dim	PTang	PRad	PRBF	PLap	PBin	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
Facebook	2	$63.3_{0.6}$	$66.6_{2.8}$	$74.3_{1.0}$	$74.8_{1.2}$	$67.5_{1.3}$	$84.3_{1.0}$	$82.0_{1.8}$	$77.5_{0.9}$	$80.1_{0.7}$	$74.1_{1.4}$	$74.4_{3.6}$	$71.9_{2.6}$	$76.0_{1.9}$	$69.4_{3.3}$	$80.6_{3.3}$	$81.8_{4.3}$	$82.9_{3.7}$	$\mathbf{85.5}_{1.2}$
	5	$65.2_{0.5}$	$81.6_{1.1}$	$87.2_{0.8}$	$87.9_{0.6}$	$86.0_{1.2}$	$89.9_{0.3}$	$89.7_{0.4}$	$88.0_{0.8}$	$88.1_{0.7}$	$86.3_{1.6}$	$83.1_{1.8}$	$60.7_{2.0}$	$67.9_{2.7}$	$79.8_{2.3}$	$87.6_{2.8}$	$88.1_{2.2}$	$88.1_{2.1}$	$\mathbf{90.5}_{0.3}$
	10	$74.0_{0.8}$	$86.5_{1.1}$	$88.2_{0.7}$	$88.5_{0.5}$	$88.4_{1.4}$	$90.6_{0.4}$	$90.6_{0.3}$	$88.4_{0.6}$	$88.8_{0.3}$	$87.0_{2.7}$	$84.3_{2.5}$	$59.0_{2.8}$	$66.8_{3.3}$	$81.4_{2.2}$	$88.0_{1.7}$	$89.5_{1.3}$	$89.6_{1.4}$	$\mathbf{91.4}_{0.8}$
	25	$85.8_{0.5}$	$88.9_{0.6}$	$88.0_{0.5}$	$88.2_{0.7}$	$89.6_{1.0}$	$90.8_{0.2}$	$90.9_{0.3}$	$88.4_{0.6}$	$88.6_{0.3}$	$87.2_{1.9}$	$85.1_{1.7}$	$58.9_{3.2}$	$64.4_{3.2}$	$83.1_{2.1}$	$88.3_{2.1}$	$90.1_{1.1}$	$90.2_{1.5}$	$\mathbf{91.5}_{0.4}$
Terrorist	2	$51.0_{2.3}$	$74.7_{3.0}$	$82.6_{1.6}$	$80.1_{1.5}$	$84.4_{0.6}$	$83.1_{2.5}$	$81.1_{1.4}$	$80.8_{2.3}$	$82.5_{2.3}$	$83.1_{3.1}$	$82.8_{2.0}$	$82.3_{3.1}$	$82.6_{3.6}$	$82.0_{2.6}$	$85.6_{2.4}$	$86.3_{3.1}$	$\mathbf{87.3}_{1.9}$	$86.4_{1.9}$
	5	$60.0_{2.2}$	$74.5_{2.8}$	$75.7_{1.8}$	$84.7_{0.9}$	$69.1_{2.3}$	$89.3_{0.9}$	$86.9_{0.5}$	$77.3_{2.2}$	$83.6_{2.1}$	$87.0_{2.1}$	$87.0_{1.8}$	$79.6_{1.0}$	$85.8_{1.5}$	$89.3_{2.8}$	$88.3_{3.2}$	$89.2_{2.2}$	$87.8_{1.3}$	$\mathbf{90.9}_{2.5}$
	10	$62.8_{0.2}$	$75.1_{3.1}$	$73.4_{1.2}$	$83.8_{1.1}$	$68.3_{1.2}$	$88.6_{1.2}$	$86.1_{2.6}$	$75.1_{2.1}$	$83.0_{1.8}$	$89.5_{1.7}$	$87.2_{1.8}$	$78.2_{2.5}$	$85.6_{2.9}$	$87.8_{2.6}$	$88.9_{2.3}$	$88.9_{2.5}$	$89.2_{2.5}$	$\mathbf{91.5}_{2.0}$
	25	$62.8_{0.2}$	$73.9_{3.2}$	$73.1_{1.9}$	$84.4_{1.2}$	$68.2_{1.4}$	$89.5_{2.3}$	$86.1_{1.8}$	$75.6_{1.6}$	$83.5_{3.2}$	$89.3_{2.0}$	$86.9_{1.6}$	$77.3_{2.6}$	$85.2_{2.6}$	$87.9_{2.1}$	$88.9_{2.0}$	$89.8_{2.7}$	$88.7_{2.3}$	$\mathbf{91.6}_{1.5}$
AC	2	$66.0_{1.2}$	$61.4_{7.0}$	$74.8_{1.3}$	$75.9_{0.8}$	$72.2_{1.5}$	$76.3_{0.9}$	$75.4_{1.1}$	$75.7_{0.8}$	$77.3_{0.8}$	$74.9_{1.5}$	$76.3_{1.4}$	$72.8_{1.4}$	$76.6_{1.2}$	$74.7_{2.5}$	$77.4_{0.5}$	$76.5_{0.8}$	$75.6_{0.7}$	$\mathbf{79.8}_{1.1}$
	5	$75.9_{1.4}$	$79.1_{0.8}$	$85.1_{1.0}$	$85.1_{1.4}$	$84.1_{0.9}$	$86.3_{0.8}$	$85.9_{0.6}$	$85.5_{0.8}$	$85.1_{1.1}$	$84.4_{0.9}$	$80.9_{0.5}$	$58.6_{3.4}$	$71.2_{1.5}$	$80.3_{1.0}$	$84.9_{1.5}$	$86.4_{1.5}$	$85.9_{1.4}$	$\mathbf{88.2}_{1.6}$
	10	$82.0_{0.6}$	$86.7_{1.1}$	$87.6_{0.7}$	$87.6_{0.5}$	$87.1_{1.2}$	$\mathbf{89.5}_{0.6}$	$88.8_{0.5}$	$87.6_{0.8}$	$88.1_{0.8}$	$85.9_{1.3}$	$84.0_{1.0}$	$58.9_{3.1}$	$72.3_{1.8}$	$83.3_{1.1}$	$88.4_{0.6}$	$88.2_{0.8}$	$87.7_{0.8}$	$\mathbf{89.5}_{1.8}$
	25	$85.4_{0.4}$	$87.7_{0.8}$	$87.1_{0.7}$	$87.0_{0.7}$	$87.5_{0.7}$	$89.6_{0.5}$	$89.3_{0.4}$	$87.5_{0.5}$	$87.8_{0.7}$	$85.8_{1.0}$	$84.7_{0.6}$	$57.4_{3.1}$	$71.0_{2.0}$	$83.9_{0.3}$	$89.4_{0.9}$	$89.3_{0.9}$	$89.3_{1.6}$	$\mathbf{90.5}_{0.6}$
Cora ML	2	$13.4_{3.0}$	$43.0_{1.2}$	$71.5_{1.2}$	$74.2_{1.6}$	$66.4_{1.0}$	$71.4_{1.6}$	$70.4_{1.5}$	$69.7_{1.1}$	$71.7_{1.2}$	$71.3_{1.5}$	$70.8_{1.7}$	$70.3_{1.5}$	$72.3_{1.6}$	$61.3_{1.2}$	$73.7_{3.3}$	$74.3_{2.7}$	$73.6_{2.2}$	$\mathbf{74.8}_{1.5}$
	5	$26.6_{0.4}$	$44.8_{1.0}$	$43.4_{0.9}$	$61.9_{0.8}$	$35.6_{0.8}$	$82.2_{2.2}$	$79.7_{1.4}$	$48.8_{1.0}$	$61.0_{1.2}$	$82.2_{2.2}$	$79.7_{1.4}$	$52.8_{1.2}$	$64.9_{1.1}$	$77.1_{0.9}$	$79.7_{1.8}$	$80.4_{1.0}$	$80.6_{1.0}$	$\mathbf{83.3}_{2.7}$
	10	$27.3_{0.4}$	$46.3_{2.0}$	$40.7_{0.9}$	$60.6_{1.3}$	$34.1_{0.4}$	$83.9_{0.8}$	$82.0_{0.7}$	$46.1_{1.1}$	$59.9_{2.1}$	$83.9_{0.8}$	$82.0_{0.7}$	$50.0_{0.8}$	$63.2_{1.3}$	$80.6_{1.2}$	$83.9_{1.3}$	$83.2_{1.6}$	$84.1_{1.1}$	$\mathbf{88.7}_{1.8}$
	25	$27.7_{0.5}$	$43.5_{1.6}$	$39.2_{0.8}$	$58.2_{1.3}$	$33.4_{0.5}$	$84.8_{1.1}$	$81.9_{1.1}$	$44.8_{1.5}$	$58.3_{1.2}$	$84.8_{1.1}$	$82.1_{1.1}$	$48.5_{1.5}$	$61.9_{1.1}$	$80.3_{0.8}$	$84.6_{0.8}$	$84.7_{0.8}$	$84.7_{1.0}$	$\mathbf{89.3}_{1.6}$
LastFM	2	$8.6_{1.6}$	$71.5_{3.1}$	$78.0_{1.0}$	$85.6_{2.7}$	$60.7_{1.7}$	$85.9_{2.9}$	$81.9_{3.0}$	$74.4_{2.9}$	$82.9_{3.5}$	$85.5_{3.9}$	$81.7_{3.3}$	$76.0_{2.8}$	$83.9_{3.0}$	$81.7_{3.0}$	$85.9_{3.7}$	$85.5_{4.0}$	$86.3_{3.3}$	$\mathbf{88.0}_{1.8}$
	5	$13.4_{0.4}$	$78.4_{2.9}$	$50.1_{1.3}$	$71.4_{2.1}$	$28.5_{2.6}$	$90.2_{2.3}$	$86.8_{1.6}$	$46.6_{2.3}$	$63.0_{2.8}$	$90.2_{2.3}$	$86.9_{1.8}$	$51.1_{2.5}$	$65.9_{3.9}$	$86.3_{1.5}$	$90.2_{3.6}$	$90.2_{4.6}$	$\mathbf{90.7}_{3.7}$	$90.3_{0.8}$
	10	$13.4_{0.4}$	$80.6_{2.4}$	$48.2_{1.9}$	$70.4_{2.5}$	$25.9_{1.7}$	$90.5_{1.8}$	$88.9_{2.1}$	$47.8_{2.8}$	$61.3_{3.2}$	$90.5_{1.8}$	$88.8_{1.7}$	$51.5_{3.1}$	$64.2_{2.7}$	$87.1_{1.8}$	$91.7_{2.4}$	$91.4_{2.3}$	$91.5_{2.2}$	$\mathbf{91.9}_{1.2}$
	25	$13.4_{0.4}$	$79.7_{3.2}$	$50.1_{0.9}$	$70.1_{2.5}$	$26.8_{2.5}$	$90.8_{1.7}$	$89.2_{1.3}$	$45.6_{2.5}$	$62.2_{4.6}$	$90.8_{1.7}$	$88.9_{1.3}$	$51.1_{3.0}$	$65.5_{4.4}$	$86.9_{0.5}$	$92.1_{2.4}$	$\mathbf{92.4}_{2.2}$	$\mathbf{92.4}_{2.5}$	$92.3_{1.7}$

Table 12: Semantic textual similarity results on STS-B benchmark. The baseline result is taken from Si et al. (2026), and the best result is highlighted in **bold**.

Method	Baseline	PTang	PRBF	PLap	PBin	PRad	CHL	CHPoly	CHRBF	CHLap	AHL	AHPoly	AHRBF	AHLap	AHRad	RIBin	RIExp	RIInt	RILog
SRC	84.24	84.36	84.84	84.63	84.06	84.53	84.70	84.63	83.98	84.74	84.27	84.68	84.33	84.48	85.16	84.92	85.53	85.45	**86.31**