

Manifold Embedding: A Point-to-Hyperplane Approach

Xianglong Shi¹, Yunhan Jiang³, Nicu Sebe², Ziheng Chen^{2*}

¹University of Science and Technology of China, ²University of Trento, ³Peking University
xlshi@mail.ustc.edu.cn, ziheng_ch@163.com

Abstract

Neural networks over non-Euclidean manifolds have attracted increasing attention across various applications. A fundamental step in these networks is to map Euclidean features onto a target manifold. Existing methods mainly use exponential maps and projections, operate through tangent or ambient Euclidean spaces, and may therefore distort intrinsic manifold geometry. On the other hand, point-to-hyperplane distances have recently shown great success in building manifold-valued networks. Based on this, we propose the Point-to-Hyperplane Distance Map (PHDM), a parameter-free mapping that extends Euclidean fully connected layers to Euclidean-to-manifold transformations. Specially, the exponential map is recovered as a pseudo-distance-based special case. We manifest PHDM on the Lorentz, Poincaré, spherical, and projected-spherical models, covering both negative and positive curvature. Experiments on arithmetic reasoning across four LLMs, image classification, graph node classification, and link prediction demonstrate consistent improvements over the corresponding baselines. The code is available at [this url](#).

1 Introduction

Neural networks over non-Euclidean manifolds have achieved remarkable success in a wide range of applications, including hierarchical representation learning, graph learning, computer vision, and Natural Language Processing (NLP) (Nickel and Kiela, 2017, 2018; Ganea et al., 2018; Chami et al., 2019; Cohen et al., 2018; Gu et al., 2019; Bachmann et al., 2020; Skopek et al., 2020; Chen et al., 2025a,c; Li et al., 2026). In NLP, non-Euclidean representations have been applied to tasks such as temporal relation extraction, keyphrase ranking, and model unlearning (Tan et al., 2021; Song et al., 2023; Wang et al., 2025). A common and fundamental step in these networks is to map Euclidean

features onto a target manifold before subsequent manifold operations.

This work focuses on this input mapping. Existing manifold neural networks often use an exponential map or a projection from the ambient space into the target manifold to produce a valid manifold point (Ganea et al., 2018; Chami et al., 2019; Liu et al., 2019; Shimizu et al., 2021; Chen et al., 2022). Meanwhile, recent work has established point-to-hyperplane distances as an effective method for constructing fully connected layers on manifolds (Chen et al., 2025b; Shi et al., 2026). Besides, the commonly used exponential map can also be interpreted by a pseudo-distance-based point-to-hyperplane formulation (Chen et al., 2024, 2025b). Building on this perspective, we study the Point-to-Hyperplane Distance Map (PHDM): a parameter-free map that interprets each input component as a signed distance to a canonical hyperplane and constructs the output point by solving the corresponding point-to-hyperplane distance equations exactly. For an input distance vector v , the target point y is defined by

$$d_{\mathcal{M}}^{\pm}(y, H_{e^{(k)},0}^{\mathcal{M}}) = v_k(x), \quad k = 1, \dots, n, \quad (1)$$

where $H_{e^{(k)},0}^{\mathcal{M}}$ is the manifold hyperplane through the origin and orthogonal to the k -th basis direction, $d_{\mathcal{M}}^{\pm}$ is the signed point-to-hyperplane distance, and $v_k(x) = \langle w_k, x \rangle - b_k$. We derive closed-form instantiations of PHDM for four constant-curvature manifolds, which are the Lorentz model, Poincaré ball, sphere, and projected sphere.

We evaluate PHDM by replacing only the input map in four experimental settings. As our primary NLP evaluation, we compare PHDM LoRA with the reported LoRA and HypLoRA rows on MAWPS, SVAMP, GSM8K, and AQuA under matched parameter budgets (Hu et al., 2022; Yang et al., 2025), across representative open-weight LLMs, including Qwen (Qwen Team, 2024),

*Corresponding author.

LLaMA (Touvron et al., 2023; Grattafiori et al., 2024), and Gemma (Gemma Team, 2024). In image classification, the HCN setting uses ResNet-18 backbones and reports CIFAR-10, CIFAR-100, and Tiny-ImageNet accuracies (Bdeir et al., 2024; He et al., 2016; Krizhevsky, 2009; Deng et al., 2009); this experiment evaluates both Poincaré and Lorentz PHDM variants. In graph node classification, the Hypformer replacement is a Lorentz-only experiment and improves Hypformer on Airport, Cora, Citeseer, and PubMed (Yang et al., 2024; Vaswani et al., 2017; Sen et al., 2008; McCallum et al., 2000). In Cora link prediction, the variants cover all four target geometries: Poincaré, Lorentz, sphere, and projected sphere; the strongest spherical variant reaches 93.09 ROC AUC, and the Lorentz and projected-sphere PHDM rows improve over their reproduced base counterparts (Katsman et al., 2023; Fawcett, 2006). These results support a conservative claim: in these settings, a localized parameter-free change to the geometric input map can improve the manifold architecture without adding extra parameters.

2 Related Work

2.1 Manifold embeddings and constant-curvature representations

Hyperbolic embeddings are widely used for hierarchical representation learning, first through the Poincaré model and then through the Lorentz model (Nickel and Kiela, 2017, 2018). Hyperbolic neural networks extended this idea from static embeddings to neural operations in curved spaces (Ganea et al., 2018). Subsequent work explored graph convolutions and graph neural networks in hyperbolic space (Chami et al., 2019; Liu et al., 2019), improved practical hyperbolic layers (Shimizu et al., 2021), and pushed toward fully hyperbolic architectures (Chen et al., 2022). Other work studies constant-curvature choices more broadly, including spherical CNNs, Riemannian residual networks, and mixed-curvature representations (Cohen et al., 2018; Katsman et al., 2023; Bachmann et al., 2020; Skopek et al., 2020). Our method is complementary to these architectures: it does not replace their manifold operations, but changes the input map that decodes Euclidean features or logits into a manifold point. Many applications of Riemannian models to NLP, such as HGRU (Tan et al., 2021), HyperRank (Song et al., 2023), HypMix (Sawhney et al., 2021), and HyperUnlearn

(Wang et al., 2025), follow a common pipeline comprising a Euclidean encoder, a Euclidean-to-manifold map, and subsequent manifold computation, but prior work has largely focused on the encoder or manifold network rather than the mapping interface, which PHDM directly addresses through a unified formulation for both hyperbolic and spherical manifolds.

2.2 Projection and optimization in manifold neural networks

Manifold neural networks must keep intermediate representations inside the chosen geometry. Existing hyperbolic and constant-curvature models therefore rely on origin exponential maps, logarithmic maps, stereographic projections, model-specific retractions, and numerical projection operators to move between Euclidean tangent vectors and valid manifold points (Ganea et al., 2018; Chami et al., 2019; Liu et al., 2019; Shimizu et al., 2021; Bachmann et al., 2020; Skopek et al., 2020; Chen et al., 2022). Recent architecture families use these ingredients inside convolutional, graph, residual, Transformer, continuous-time, and parameter-efficient fine-tuning models (Bdeir et al., 2024; Katsman et al., 2023; Yang et al., 2024; Lou et al., 2020; Yang et al., 2025). Riemannian optimization methods provide a complementary component by adapting parameter updates to curved spaces (Bonnabel, 2013; Bécigneul and Ganea, 2019).

These tools are usually designed for geometric validity, local tangent-space parameterization, or training stability. PHDM targets a different property: component-wise signed-distance preservation with respect to canonical hyperplanes. A projection operator can repair an invalid point, but it does not generally preserve the signed distances $d_{\mathcal{M}}^{\pm}(y, H_{e^{(k)},0}^{\mathcal{M}})$. An origin exponential map follows the radial geodesic determined by a tangent vector, but it does not generally solve the component-wise distance equations introduced below. This distinction matters because PHDM gives each scalar v_k an explicit geometric meaning as a signed distance rather than treating it as an unconstrained tangent-vector component.

2.3 Point-to-hyperplane semantics versus learned hyperplanes

The point-to-hyperplane view has appeared mainly in hyperbolic classifiers and transformation layers. HNN formulates multinomial logistic regression

through learned Poincaré hyperplanes, where class scores are signed margins to geodesic decision boundaries (Ganea et al., 2018). HNN++ further uses this interpretation to build Poincaré fully connected layers: a Euclidean FC output is treated as a signed distance to an output hyperplane, and the layer reconstructs a Poincaré point whose distances match those scores (Shimizu et al., 2021). These works use hyperplanes as trainable classifiers or transformation-layer objects. PHDM instead fixes the output-side hyperplanes canonically and uses the point-to-hyperplane equations as a parameter-free input map from Euclidean scores into the target manifold. This makes PHDM complementary to the systems used in our experiments, including HypLoRA (Yang et al., 2025), HCNNet-style image-classification baselines (Bdeir et al., 2024), Hypformer (Yang et al., 2024), and RResNet (Katsman et al., 2023), which provide the surrounding task model while PHDM changes only the input map.

3 Preliminaries

We organize the target spaces through two constant-curvature model families rather than four unrelated model names. The *radius model* $\mathcal{M}_K^n$ unifies the Lorentz hyperboloid $\mathbb{L}_K^n$ for $K < 0$ and the ambient sphere $\mathbb{S}_K^n$ for $K > 0$ (Bachmann et al., 2020; Skopek et al., 2020). The *stereographic model* $\mathfrak{st}_K^n$ unifies the Poincaré ball $\mathbb{P}_K^n$ for $K < 0$ and the projected sphere $\mathbb{D}_K^n$ for $K > 0$. We write $x = [x_0, x_s]$ for radius-model points, where x_0 is the time coordinate and $x_s \in \mathbb{R}^n$ is the spatial part. The two model families are defined as

$$\mathcal{M}_K^n = \{x \in \mathbb{R}^{n+1} \mid \langle x, x \rangle_K = 1/K\}. \quad (2)$$

$$\begin{aligned} \mathfrak{st}_K^n &= \{x \in \mathbb{R}^n \mid -K\|x\|_2^2 < 1\}, \\ g_x^K &= (\lambda_x^K)^2 g^E, \quad \lambda_x^K = \frac{2}{1 + K\|x\|_2^2}. \end{aligned} \quad (3)$$

Here, $g^E = \langle \cdot, \cdot \rangle$ denotes the standard Euclidean metric. For $K < 0$, $\mathcal{M}_K^n$ denotes the future sheet with $x_0 > 0$. The curvature-aware functions and inner product are

$$\begin{aligned} \sin_K(t) &= \begin{cases} \sinh(t), & K < 0, \\ \sin(t), & K > 0, \end{cases} & \cos_K(t) &= \begin{cases} \cosh(t), & K < 0, \\ \cos(t), & K > 0, \end{cases} \\ \tan_K(t) &= \begin{cases} \tanh(t), & K < 0, \\ \tan(t), & K > 0, \end{cases} & \langle u, v \rangle_K &= \begin{cases} \langle u, v \rangle_{\mathcal{L}}, & K < 0, \\ \langle u, v \rangle, & K > 0. \end{cases} \end{aligned}$$

For stereographic models, $u = (-x) \oplus_K y$, and $\oplus_K$ is the standard signed-curvature Mobius addition (Skopek et al., 2020; Ganea et al., 2018; Nickel and Kiela, 2018).

4 Methodology

This section derives PHDM from the transformation view of fully connected layers, using true point-to-hyperplane distances. We first revisit the Euclidean score and its point-to-hyperplane interpretation, then using the same principle, we define PHDM as a parameter-free Euclidean-to-manifold input map that preserves signed distances to canonical output hyperplanes. We then express the construction through the two model families defined above: the radius model $\mathcal{M}_K^n$ and the stereographic model $\mathfrak{st}_K^n$. In each case, we first define the canonical hyperplanes and their signed point-to-hyperplane distances, and then solve the resulting distance constraints to obtain a closed-form PHDM. Furthermore, as detailed in the Appendix (Appendix E), the commonly used origin exponential map can be interpreted as a pseudo-distance-based construction because it reconstructs tangent-space pseudo-distances (Chen et al., 2024, 2025b). In contrast, PHDM reconstructs manifold points from true signed point-to-hyperplane distances.

4.1 Euclidean and Riemannian transformations

The Euclidean fully connected layer is defined as $y = Wx + b$ with $W \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$. Shimizu et al. (Shimizu et al., 2021, Sec. 3.2) interpret Euclidean FC layers through signed distances: we write each output component as $v_k(x) = \langle w_k, x \rangle - b_k$ and view it as the signed distance from y to the axis-aligned hyperplane

$$\begin{aligned} H_{e^{(k)},0} &= \{y \in \mathbb{R}^n \mid \langle e^{(k)}, y \rangle = 0\}, \\ \text{sign}(\langle e^{(k)}, y \rangle) d(y, H_{e^{(k)},0}) &= v_k(x). \end{aligned} \quad (4)$$

where $e^{(k)}$ is the k th standard basis vector. The left-hand side of Equation (4) is exactly the k th output component.

Equation (4) shows that a Euclidean FC layer can be read as a set of signed-distance constraints, one constraint for each output component. PHDM keeps the scalar scores $v_k(x)$ from this Euclidean layer but changes the space in which the constraints are satisfied. Instead of reconstructing a Euclidean vector whose k th coordinate equals $v_k(x)$, PHDM reconstructs a manifold point $Y \in \mathcal{M}$ whose signed distance to the canonical manifold hyperplane $H_{e^{(k)},0}^{\mathcal{M}}$ equals the same scalar:

$$d_{\mathcal{M}}^{\pm}(Y, H_{e^{(k)},0}^{\mathcal{M}}) = v_k(x), \quad k = 1, \dots, n. \quad (5)$$

Table 1: Two constant-curvature model families used in this paper. The stereographic model $\mathfrak{st}_K^n$ gives the Poincaré ball for $K < 0$ and the projected sphere for $K > 0$. The radius model $\mathcal{M}_K^n$ gives the Lorentz hyperboloid for $K < 0$ and the ambient sphere for $K > 0$.

	Distance $d(x, y)$	Exponential map $\exp_x(v)$	Logarithmic map $\log_x(y)$
$\mathfrak{st}_K^n$	$\frac{2}{\sqrt{ K }} \tan^{-1} \left(\sqrt{ K } \ u\ _2 \right)$	$x \oplus_K \left(\tan_K \left(\frac{\sqrt{ K } \lambda_x^K \ v\ _2}{2} \right) \frac{v}{\sqrt{ K } \ v\ _2} \right)$	$\frac{2}{\sqrt{ K } \lambda_x^K} \tan^{-1} \left(\sqrt{ K } \ u\ _2 \right) \frac{u}{\ u\ _2}$
$\mathcal{M}_K^n$	$\frac{1}{\sqrt{ K }} \cos_K^{-1} (K \langle x, y \rangle_K)$	$\cos_K(\sqrt{ K } \ v\ _K) x + \frac{\sin_K(\sqrt{ K } \ v\ _K)}{\sqrt{ K } \ v\ _K} v$	$\frac{\cos_K^{-1}(K \langle x, y \rangle_K)}{\sin_K(\cos_K^{-1}(K \langle x, y \rangle_K))} (y - K \langle x, y \rangle_K x)$

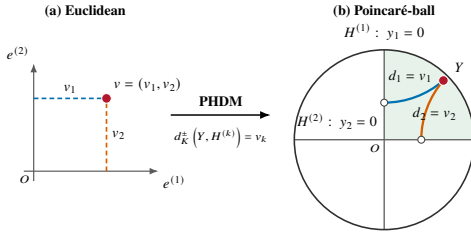


Figure 1: Two-dimensional illustration of PHDM in the Poincaré ball. The Euclidean components v_1 and v_2 become signed geodesic distances from Y to the canonical hyperplanes $H^{(1)}$ and $H^{(2)}$.

Thus the Euclidean scores are not treated as tangent-vector coordinates. They are prescribed signed distances to fixed output hyperplanes, and the output point is obtained by solving the corresponding n point-to-hyperplane equations. Figure 1 illustrates this construction in two dimensions for the Poincaré ball.

The closed forms below share the same scalar inversion step. For curvature $K < 0$, define $r_K(v)$ elementwise by

$$r_K(v) = (-K)^{-1/2} \sinh(\sqrt{-K}v). \quad (6)$$

For curvature $K > 0$, define $r_K(v)$ elementwise by

$$r_K(v) = K^{-1/2} \sin(\sqrt{K}v). \quad (7)$$

The remaining step is model-family reconstruction: radius models recover the missing time coordinate, while stereographic models apply the corresponding stereographic reconstruction.

4.2 PHDM on the radius model

In the radius model, the canonical hyperplanes are ambient coordinate hyperplanes. We first detail the Lorentz case. For $K < 0$, $\mathcal{M}_K^n$ is the Lorentz hyperboloid $\mathbb{L}_K^n$. Following Bdeir et al. (2024), a Lorentz hyperplane passing through $p \in \mathbb{L}_K^n$ and

perpendicular to $w \in T_p \mathbb{L}_K^n$ is the intersection of the hyperboloid with an ambient linear hyperplane:

$$H_{w,p}^{\mathbb{L}_K} = \{x \in \mathbb{L}_K^n \mid \langle w, x \rangle_{\mathcal{L}} = 0\}. \quad (8)$$

Under the Euclidean reparameterization used by Bdeir et al. (2024), the corresponding point-to-hyperplane distance for the reparameterized hyperplane $\tilde{H}_{z,a}^{\mathbb{L}_K}$, with $x = [x_t, x_s] \in \mathbb{L}_K^n$, $z \in \mathbb{R}^n$, and $a \in \mathbb{R}$, can be written as

$$d_{\mathbb{L}_K}(x, \tilde{H}_{z,a}^{\mathbb{L}_K}) = \frac{1}{\sqrt{-K}} \left| \operatorname{arsinh} \left(\sqrt{-K} \frac{\cosh(\sqrt{-K}a) \langle z, x_s \rangle - \sinh(\sqrt{-K}a) \|z\|_2 x_t}{\sqrt{\| \cosh(\sqrt{-K}a) z \|^2_2 - (\sinh(\sqrt{-K}a) \|z\|_2)^2}} \right) \right|.$$

PHDM uses the fixed canonical special case rather than learned hyperplanes. For the k -th spatial axis, the canonical hyperplane is

$$H_{e^{(k)},0}^{\mathbb{L}_K} = \{y \in \mathbb{L}_K^n \mid \langle e^{(k)}, y \rangle_K = 0\}, \quad (9)$$

and following Bdeir et al. (2024), its signed point-to-hyperplane distance is

$$d_{\mathbb{L}_K}^{\pm}(y, H_{e^{(k)},0}^{\mathbb{L}_K}) = \frac{1}{\sqrt{-K}} \operatorname{arsinh}(\sqrt{-K} y_k). \quad (10)$$

Theorem 1 (Lorentz PHDM) For any $v \in \mathbb{R}^n$ and $K < 0$, the unique future-sheet point satisfying $d_{\mathbb{L}_K}^{\pm}(y, H_{e^{(k)},0}^{\mathbb{L}_K}) = v_k$ for all $k = 1, \dots, n$ is

$$\text{PHDM}_{\mathbb{L}_K}(v) = \left[\sqrt{-\frac{1}{K} + \|r_K(v)\|_2^2}, r_K(v) \right]. \quad (11)$$

Theorem 2 (Spherical hyperplane distance)

For $K > 0$, $\mathcal{M}_K^n$ is the ambient sphere $\mathbb{S}_K^n$. For the k -th ambient axis, define

$$H_{e^{(k)},0}^{\mathbb{S}_K} = \{y \in \mathbb{S}_K^n \mid \langle e^{(k)}, y \rangle_K = 0\}. \quad (12)$$

On the principal branch, its signed point-to-hyperplane distance is

$$d_{\mathbb{S}_K}^{\pm}(y, H_{e^{(k)},0}^{\mathbb{S}_K}) = \frac{1}{\sqrt{K}} \arcsin(\sqrt{K} y_k). \quad (13)$$

Theorem 3 (Sphere PHDM) For $K > 0$, if v satisfies the positive-curvature feasibility conditions, the north-hemisphere point satisfying $d_{\mathbb{S}_K}^\pm(y, H_{e^{(k)},0}^{\mathbb{S}_K}) = v_k$ for all $k = 1, \dots, n$ is

$$\text{PHDM}_{\mathbb{S}_K}(v) = \left[\sqrt{\frac{1}{K} - \|r_K(v)\|_2^2}, r_K(v) \right]. \quad (14)$$

Here, the feasibility conditions are in Appendix Equation (58). The radius-model formula therefore has the same structure in both curvature regimes: invert the signed distance into the spatial vector $r_K(v)$, then recover the remaining coordinate from the constraint of $\mathcal{M}_K^n$.

4.3 PHDM on the stereographic model

In the stereographic model, the canonical hyperplanes are coordinate hyperplanes. For $K < 0$, $\mathfrak{st}_K^n$ is the Poincaré ball $\mathbb{P}_K^n$. For the k -th axis, the canonical hyperplane is

$$H_{e^{(k)},0}^{\mathbb{P}_K} = \{y \in \mathbb{P}_K^n \mid \langle e^{(k)}, y \rangle = 0\}, \quad (15)$$

Following Shimizu et al. (2021), its signed point-to-hyperplane distance is

$$d_{\mathbb{P}_K}^\pm(y, H_{e^{(k)},0}^{\mathbb{P}_K}) = \frac{1}{\sqrt{-K}} \operatorname{arsinh} \left(\frac{2\sqrt{-K}y_k}{1 + K\|y\|_2^2} \right). \quad (16)$$

Theorem 4 (Poincaré PHDM) For any $v \in \mathbb{R}^n$ and $K < 0$, the point satisfying $d_{\mathbb{P}_K}^\pm(y, H_{e^{(k)},0}^{\mathbb{P}_K}) = v_k$ for all $k = 1, \dots, n$ is

$$\text{PHDM}_{\mathbb{P}_K}(v) = \frac{r_K(v)}{1 + \sqrt{1 - K\|r_K(v)\|_2^2}}. \quad (17)$$

Theorem 5 (Projected-sphere distance) For $K > 0$, $\mathfrak{st}_K^n$ is the projected sphere $\mathbb{D}_K^n$. For axis k in this model, define

$$H_{e^{(k)},0}^{\mathbb{D}_K} = \{y \in \mathbb{D}_K^n \mid \langle e^{(k)}, y \rangle = 0\}. \quad (18)$$

On the principal branch, its signed point-to-hyperplane distance is

$$d_{\mathbb{D}_K}^\pm(y, H_{e^{(k)},0}^{\mathbb{D}_K}) = \frac{1}{\sqrt{K}} \arcsin \left(\frac{2\sqrt{K}y_k}{1 + K\|y\|_2^2} \right).$$

Theorem 6 (Projected-sphere PHDM) For $K > 0$, if v satisfies the positive-curvature feasibility conditions, the north-branch projected-sphere point satisfying $d_{\mathbb{D}_K}^\pm(y, H_{e^{(k)},0}^{\mathbb{D}_K}) = v_k$ for all $k = 1, \dots, n$ is

$$\text{PHDM}_{\mathbb{D}_K}(v) = \frac{r_K(v)}{1 + \sqrt{1 - K\|r_K(v)\|_2^2}}. \quad (19)$$

4.4 Unified view

Parameterization. PHDM separates trainable score generation from geometric decoding. The score vector $v(x)$ can be produced by any Euclidean module, such as a fully connected layer $v(x) = Wx + b$ or a parameter-efficient update inside a larger architecture. These trainable parameters belong to the upstream model. Given $v(x)$, PHDM fixes the target curvature K and the canonical hyperplanes $H_{e^{(k)},0}^{\mathcal{M}}$, and reconstructs the manifold point by solving Eq. (5). Therefore, PHDM introduces no learned hyperplane normal, offset, or bias, and replacing an input map with PHDM does not change the trainable parameter count.

Table 2 places the four constructions in the same signed-distance template but groups them by model family. This view separates PHDM from three common operations. A projection operator enforces domain membership but does not preserve signed point-to-hyperplane distances. An origin exponential map follows a radial geodesic and therefore does not generally solve Eq. (5). A learned hyperplane classifier changes the normal or offset of the hyperplane, whereas PHDM fixes $H_{e^{(k)},0}^{\mathcal{M}}$ and treats the point-to-hyperplane equations themselves as the definition of the output point.

Comparison to HNN++. Both HNN++ and PHDM reconstruct a manifold point from signed distances to hyperplanes, but this reconstruction serves a different role in each method. HNN++ uses it within a learned Poincaré-to-Poincaré transformation, $\mathbb{P}_K^n \rightarrow \mathbb{P}_K^n$, whose MLR and fully connected parameters define the transformation (Shimizu et al., 2021). PHDM instead fixes the canonical hyperplanes and provides a parameter-free input map from Euclidean scores, namely $\mathbb{R}^n \rightarrow \mathbb{P}_K^n$, $\mathbb{R}^n \rightarrow \mathbb{L}_K^n$, $\mathbb{R}^n \rightarrow \mathbb{S}_K^n$, or $\mathbb{R}^n \rightarrow \mathbb{D}_K^n$. In our experiments, PHDM replaces only the original origin exponential map or architecture-specific input projection while leaving the surrounding encoder and manifold architecture unchanged. Our contribution is therefore the Euclidean-to-manifold input interface rather than a new Poincaré fully connected or MLR layer.

5 Experiments

5.1 Numerical stability

Following Chen et al. (2026), we compare PHDM with the standard origin exponential/logarithmic round trip. The benchmark samples 4096 vectors in dimension 16, uses signed curvature $K = -1$

Table 2: Signed point-to-hyperplane distances and PHDM formulas under the two unified constant-curvature model families. Here $r_K = r_K(v)$, with $r_K(v) = (-K)^{-1/2} \sinh(\sqrt{-K}v)$ for $K < 0$ and $r_K(v) = K^{-1/2} \sin(\sqrt{K}v)$ for $K > 0$, and $\text{sgn}(K)$ denotes the sign of K .

Family	Curvature	Instance	Point-to-hyperplane distance	PHDM formula
$\mathcal{M}_K^n$	$K < 0$	Lorentz	$\frac{1}{\sqrt{ K }} \sin_K^{-1}(\sqrt{ K }y_k)$	$\left[\sqrt{\frac{1}{ K } - \text{sgn}(K)\ r_K\ _2^2}, r_K \right]$
	$K > 0$	Sphere		
$\mathfrak{st}_K^n$	$K < 0$	Poincaré	$\frac{1}{\sqrt{ K }} \sin_K^{-1}\left(\frac{2\sqrt{ K }y_k}{1+K\ y\ _2^2}\right)$	$\frac{r_K}{1+\sqrt{1-K\ r_K\ _2^2}}$
	$K > 0$	Projected sphere		

or $K = +1$, and measures whether the input value is recovered after mapping to the manifold and back. For negative curvature, random-radius sampling is appropriate because the Lorentz/Poincaré PHDM variants are globally defined up to floating-point overflow. For positive curvature, arbitrary-radius tangent sampling is not a valid PHDM stress test: the spherical signed-distance values are only defined on the compact feasible set $\sum_k \sin^2(\sqrt{K}v_k) \leq 1$ and $|\sqrt{K}v_k| \leq \pi/2$ for all k . We therefore sample positive-curvature PHDM by controlling $\rho = \|\sin(\sqrt{K}v)\|_2$, where $\rho < 1$ is feasible and $\rho \rightarrow 1$ approaches the boundary.

Table 3 shows that Lorentz PHDM preserves signed-distance values to FP32 precision through the medium-radius regime, whereas the implemented Exp / Log remains finite but no longer preserves the input tangent component because the implementation clips large tangent norms. The Poincaré ball becomes ill-conditioned near its boundary for both PHDM and the Exp / Log. The Poincaré failures in the table are therefore numerical boundary or overflow failures at the large-radius test, not mathematical violations of the negative-curvature PHDM domain. For positive curvature, feasible-domain sampling is the appropriate stability test: at $\rho = 0.99$, both spherical PHDM variants remain feasible for all samples and recover signed-distance values to FP32 relative error around 10^{-8} . By contrast, random radius-2 samples in dimension 16 are outside the joint feasible domain, so they evaluate mathematical domain coverage rather than numerical stability.

5.2 Comparative experiment

All comparative experiments evaluate the same localized intervention: replace the original origin exponential map or architecture-specific input projection with the PHDM, while keeping the remaining architecture fixed. PHDM is not a learned layer and

does not introduce any extra parameters. Results are reported as means and standard deviations over five runs. LLM arithmetic reasoning and image classification use accuracy, graph node classification uses F1-score or accuracy depending on the dataset, and Cora link prediction uses ROC AUC. All experiments are conducted on A100 64G GPUs.

Image classification. Following the HCNN setting (Bdeir et al., 2024) on CIFAR-10, CIFAR-100, and Tiny-ImageNet (Bdeir et al., 2024; He et al., 2016; Krizhevsky, 2009; Deng et al., 2009), we replace only the origin exponential-map input map in Hybrid Poincaré and Hybrid Lorentz with PHDM. Table 4 shows consistent gains across all datasets: Poincaré PHDM improves by +0.42, +1.77, and +1.24 points, while Lorentz PHDM improves by +0.46, +0.52, and +0.58. The PHDM rows also achieve the best mean accuracy on every dataset and reduce standard deviation in five of the six replacements.

Graph node classification. The graph node-classification experiment uses the Hypformer setting (Yang et al., 2024; Vaswani et al., 2017). Table 5 adds the original Hypformer benchmark rows and our controlled replacement, which keeps Hypformer fixed and replaces only the Lorentz Euclidean-to-manifold input map with PHDM. In our controlled replacement experiments, PHDM improves Hypformer by +1.03, +0.68, +1.10, and +1.22 points on Airport, Cora, Citeseer, and PubMed, respectively. Its standard deviation is lower on Airport, Cora, and Citeseer and comparable on PubMed.

Cora link prediction. The Cora link-prediction experiments use Riemannian residual baselines (Katsman et al., 2023; Sen et al., 2008; McCallum et al., 2000). Table 6 reports ROC AUC in percent. PHDM improves every paired row, with gains of 0.99–3.07 points for RResNet and 0.54–1.60 for RResNet Graph. The sphere gives the best AUC

Table 3: FP32 operator stability for PHDM and $\text{Exp}_0 / \text{Log}_0$ maps. Bad results are in red. The last column distinguishes numerical failures in negative curvature from mathematical domain violations in positive curvature.

Map	Geometry	Sampling	Rel. error	Numerical failure / domain violation
PHDM	Lorentz	random $r = 10/50$	6.02×10^{-9}	0.00%
$\text{Exp}_0 / \text{Log}_0$	Lorentz	random $r = 10/50$	5.39×10^{-1}	0.00%
PHDM	Poincaré	random $r = 10/50$	1.94×10^{-6}	54.10%
$\text{Exp}_0 / \text{Log}_0$	Poincaré	random $r = 10/50$	1.36×10^{-1}	65.41%
PHDM	Sphere	feasible $\rho = 0.99$	1.46×10^{-8}	0.00%
$\text{Exp}_0 / \text{Log}_0$	Sphere	feasible $\rho = 0.99$	6.28×10^{-8}	0.00%
PHDM	Projected sphere	feasible $\rho = 0.99$	4.85×10^{-8}	0.00%
$\text{Exp}_0 / \text{Log}_0$	Projected sphere	feasible $\rho = 0.99$	5.50×10^{-8}	0.00%

Table 4: Classification accuracy in percent of ResNet-18 models. We estimate the mean and standard deviation from five runs. The best performance is highlighted in bold (higher is better). δ_{rel} denotes hyperbolicity

	CIFAR-10 ($\delta_{rel} = 0.26$)	CIFAR-100 ($\delta_{rel} = 0.23$)	Tiny-ImageNet ($\delta_{rel} = 0.20$)
Euclidean (He et al., 2016)	95.14 $\pm$ 0.12	77.72 $\pm$ 0.15	65.19 $\pm$ 0.12
Poincaré ResNet (van Spengler et al., 2023)	94.51 $\pm$ 0.15	76.60 $\pm$ 0.32	62.01 $\pm$ 0.56
HECNN Lorentz (Bdeir et al., 2024)	95.16 $\pm$ 0.11	78.76 $\pm$ 0.24	65.96 $\pm$ 0.18
HCCNN Lorentz (Bdeir et al., 2024)	95.14 $\pm$ 0.08	78.07 $\pm$ 0.17	65.71 $\pm$ 0.13
Hybrid Poincaré (Guo et al., 2022)	95.04 $\pm$ 0.13	77.19 $\pm$ 0.50	64.93 $\pm$ 0.38
Hybrid Poincaré + PHDM (Ours)	95.46$\pm$0.11	78.96$\pm$0.14	66.17$\pm$0.09
Hybrid Lorentz (Bdeir et al., 2024)	94.98 $\pm$ 0.12	78.03 $\pm$ 0.21	65.63 $\pm$ 0.10
Hybrid Lorentz + PHDM (Ours)	95.44$\pm$0.07	78.55$\pm$0.05	66.21$\pm$0.20

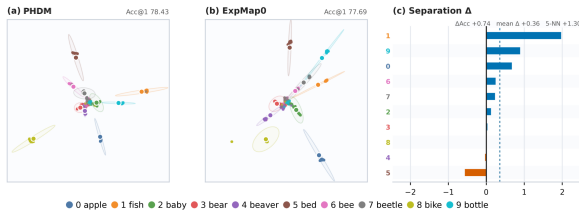


Figure 2: Lorentz CIFAR-100 test embeddings for ten fine classes. Figure (a) and (b) show PHDM and ExpMap0 embeddings under the same PCA-UMAP visualization basis (Pearson, 1901; McInnes et al., 2018). The figure reports the per-class high-dimensional separation change $\Delta_c = s_c^{\text{PHDM}} - s_c^{\text{ExpMap0}}$, where s_c is the nearest class-center distance divided by the mean within-class radius. Positive values favor PHDM.

(93.09 $\pm$ 0.44), and the projected sphere gives the largest paired gain.

LLMs. In the LoRA adapter, PHDM maps Euclidean features to the manifold, while the manifold output y is mapped back to the Euclidean residual stream via the MLR point-to-hyperplane scores $v_k^{\text{out}} = d_{\mathcal{M}}^{\perp}(y, H_{e(k),0}^{\mathcal{M}})$ (Chen et al., 2024). Table 7 evaluates whether PHDM can be used for large-scale models. We follow the LoRA/HypLoRA arithmetic-reasoning setup (Hu et al., 2022; Yang et al., 2025) and report PHDM

LoRA under the same parameter budgets as the corresponding LoRA and HypLoRA rows. Across the four active model groups, PHDM LoRA obtains the best average in every group, improving over HypLoRA by +1.29, +4.57, +0.26, and +1.21 points on LLaMA-13B, Gemma-7B, LLaMA3-8B, and Qwen2.5-7B, respectively. At the dataset level, PHDM LoRA is best in 12 of 16 cells. The strongest gain appears on Gemma-7B, where PHDM LoRA improves SVAMP, GSM8K, and especially AQuA, with a +23.99 point AQuA gain over HypLoRA. Qwen2.5-7B is the most consistent case: PHDM LoRA is best on all four datasets, supporting the effectiveness of PHDM.

Visualization. Figure 2 complements the accuracy comparison by checking whether the Lorentz PHDM gain is reflected in embedding geometry. The UMAP visualization (McInnes et al., 2018) shown in Figure 2 (a) and (b) provides a shared two-dimensional view for the selected CIFAR-100 classes, while the quantitative results shown in Figure 2 (c) are computed in the original 512-dimensional tangent feature space and are independent of the UMAP projection. PHDM has higher CIFAR-100 test accuracy than ExpMap0 by +0.74 points, improves the average separation ratio on these classes by +0.36, and improves 5-NN (Cover

Table 5: Graph node classification: F1-score for Airport and accuracy for Cora, Citeseer, and PubMed. Baselines are from Hypformer. PHDM replaces only the Euclidean-to-manifold input map.

Model	Airport	Cora	Citeseer	PubMed
HGCN (Chami et al., 2019)	89.3 $\pm$ 1.2	76.5 $\pm$ 0.6	68.0 $\pm$ 0.6	78.0 $\pm$ 1.0
HGAT (Zhang et al., 2021)	89.6 $\pm$ 1.0	77.4 $\pm$ 0.7	68.6 $\pm$ 0.3	78.3 $\pm$ 1.4
HAN (Gulcehre et al., 2019)	92.9 $\pm$ 0.6	83.1 $\pm$ 0.5	72.4 $\pm$ 0.5	79.0 $\pm$ 0.6
HNN++ (Shimizu et al., 2021)	92.3 $\pm$ 0.3	82.8 $\pm$ 0.6	71.5 $\pm$ 1.3	79.9 $\pm$ 0.4
F-HNN (Chen et al., 2022)	93.0 $\pm$ 0.7	81.0 $\pm$ 0.7	71.2 $\pm$ 0.4	77.5 $\pm$ 0.8
SGFormer (Wu et al., 2023)	92.9 $\pm$ 0.5	83.2 $\pm$ 0.9	72.2 $\pm$ 0.3	80.0 $\pm$ 0.8
Hypformer (Yang et al., 2024)	95.0 $\pm$ 0.5	85.0 $\pm$ 0.3	73.3 $\pm$ 0.4	81.3 $\pm$ 0.3
Hypformer + PHDM (Ours)	96.03$\pm$0.34	85.68$\pm$0.19	74.40$\pm$0.20	82.52$\pm$0.33

Table 6: Cora link-prediction ROC AUC in percent, following RResNet. PHDM replaces only the Euclidean-to-manifold input map, while the RResNet/RResNet Graph architectures and evaluation protocol are kept fixed.

Model	Lorentz	Poincaré	Sphere	Projected sphere
RResNet	90.68 $\pm$ 0.88	88.90 $\pm$ 0.20	90.70 $\pm$ 1.00	86.63 $\pm$ 3.37
RResNet+PHDM	91.67$\pm$0.15	91.42$\pm$0.45	93.09$\pm$0.44	89.70$\pm$0.52
RResNet Graph	90.12 $\pm$ 1.09	87.60 $\pm$ 0.90	91.70 $\pm$ 0.40	86.89 $\pm$ 0.92
RResNet Graph+PHDM	91.32$\pm$0.75	88.67$\pm$1.29	92.24$\pm$0.29	88.49$\pm$0.50

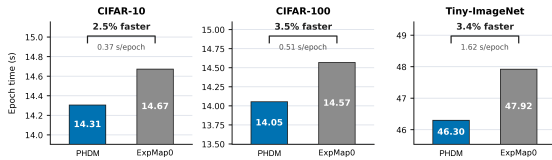


Figure 3: Per-epoch runtime comparison for the Lorentz image-classification replacement. PHDM replaces the origin exponential-map input map while keeping the surrounding model fixed.

and Hart, 1967) accuracy after PCA-50 by +1.30 points. The per-class bars show that the improvement is heterogeneous rather than universal: several classes gain substantially, while a small number favor ExpMap0.

5.3 Speed comparison

Figure 3 shows that the closed-form PHDM has lower per-epoch runtime than the origin exponential-map input map in the Lorentz image-classification setting. PHDM reduces per-epoch time from 14.67 to 14.31 seconds on CIFAR-10, from 14.57 to 14.05 seconds on CIFAR-100, and from 47.92 to 46.30 seconds on Tiny-ImageNet, corresponding to 2.5%, 3.5%, and 3.4% speedups. Thus the accuracy gains in Table 4 do not incur additional runtime cost, and replacing the input map also modestly improves training throughput.

6 Discussion

Effect of PHDM. The main benefit of PHDM is not additional capacity, but a change in input semantics. Each scalar component of the input to PHDM is interpreted as a signed distance to a hyperplane, and the output point is reconstructed to satisfy that interpretation. This makes the input map auditable: within the PHDM feasible domain, one can directly check Eq. (5). The empirical pattern suggests that this local geometric change can matter even when the larger architecture is unchanged.

PHDM-domain constraints. PHDM has different feasible domains in negative and positive curvature. Lorentz and Poincaré PHDM variants are globally defined for finite signed-distance values, with numerical care needed only for very large $\sqrt{-K}v$ values in the sinh computation when $K < 0$. Thus the Poincaré failures in Table 3 should be read as finite-precision boundary failures rather than violations of the mathematical PHDM domain. Sphere and projected-sphere PHDM variants satisfy the point-to-hyperplane equations only on the principal feasible branch satisfying the component-wise and joint feasibility conditions. In practice, clipping or radial scaling (Guo et al., 2022) provides a practical mechanism for controlling numerical stability when PHDM inputs approach or leave this feasible branch.

Table 7: Arithmetic reasoning accuracy of large language models. LoRA/HypLoRA results are reported by HypLoRA (Hu et al., 2022; Yang et al., 2025). Base model references are LLaMA (Touvron et al., 2023), Gemma (Gemma Team, 2024), Llama 3 (Grattafiori et al., 2024), and Qwen2.5 (Qwen Team, 2024). The benchmarks are MAWPS, SVAMP, GSM8K, and AQuA (Koncel-Kedziorski et al., 2016; Patel et al., 2021; Cobbe et al., 2021; Ling et al., 2017). M.AVG denotes micro-average accuracy because the datasets are imbalanced.

Base model	PEFT method	Params (%)	MAWPS (8.5%)	SVAMP (35.6%)	GSM8K (46.9%)	AQuA (9.0%)	M.AVG
LLaMA-13B	LoRA	0.67	83.5	54.7	48.5	18.5	51.0
LLaMA-13B	HypLoRA	0.67	83.2	54.8	49.0	21.5	51.5
LLaMA-13B	PHDM LoRA	0.67	84.45	61.80	46.02	22.83	52.79
Gemma-7B	LoRA	0.79	91.6	76.2	66.3	28.9	68.6
Gemma-7B	HypLoRA	0.79	89.5	78.7	69.5	32.7	71.2
Gemma-7B	PHDM LoRA	0.79	90.34	79.00	74.37	56.69	75.77
LLaMA3-8B	LoRA	0.70	92.7	78.9	70.8	30.4	71.9
LLaMA3-8B	HypLoRA	0.70	91.6	80.5	74.0	34.2	74.2
LLaMA3-8B	PHDM LoRA	0.70	91.60	78.00	76.35	34.65	74.46
Qwen2.5-7B	LoRA	0.71	90.8	84.4	78.6	68.1	80.8
Qwen2.5-7B	HypLoRA	0.71	91.2	92.2	87.9	71.6	88.3
Qwen2.5-7B	PHDM LoRA	0.71	92.86	93.50	88.78	74.41	89.51

7 Conclusion

We proposed the Point-to-Hyperplane Distance Map (PHDM) for manifold embedding. The method interprets input components as canonical signed distances and reconstructs target manifold points by solving point-to-hyperplane equations, yielding closed forms for Lorentz, Poincaré, sphere, and projected sphere geometries. In the experiments, this parameter-free input-map replacement improves several large language model, image, graph, and link-prediction baselines without adding learned PHDM parameters. The next empirical step is to extend the same controlled replacement to broader, large-scale settings in more specific areas.

8 Limitations

PHDM is an input-map replacement rather than a full architecture. The current evidence is limited to controlled replacement experiments on large language model arithmetic reasoning, image classification, graph node classification, and Cora link prediction. Positive-curvature variants are exact only on the principal feasible domain, and engineering safeguards such as clipping or radial scaling do not preserve exact signed distances.

PHDM does not introduce new data collection, user-facing decision systems, or task-specific deployment mechanisms. Its broader impact is therefore mainly methodological: it may help make manifold input maps more interpretable and easier to audit. Any downstream risks depend on the surrounding model and dataset. Practitioners should follow feasibility, handling, preprocessing, seeds,

and evaluation protocols when using PHDM in applied settings.

Acknowledgments

This work was supported by the EU Horizon projects ELIAS (No. 101120237) and ELLIOT (No. 101214398), and by the FIS project GUIDANCE (Debugging Computer Vision Models via Controlled Cross-modal Generation) (No. FIS2023-03251). We acknowledge CINECA for awarding high-performance computing resources under the ISCRA initiative and the EuroHPC Joint Undertaking for granting access to the Leonardo supercomputer at CINECA, Italy.

References

- Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. 2020. Constant curvature graph convolutional networks. In *ICML*.
- Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. 2024. Fully hyperbolic convolutional neural networks for computer vision. In *The Twelfth International Conference on Learning Representations*.
- Gary Bécigneul and Octavian-Eugen Ganea. 2019. Riemannian adaptive optimization methods. In *ICLR*.
- Silvère Bonnabel. 2013. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229.
- Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *NeurIPS*.

- Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5672–5686.
- Ziheng Chen, Yue Song, Rui Wang, Xiao-Jun Wu, and Nicu Sebe. 2024. Rmlr: Extending multinomial logistic regression into general geometries. *Advances in Neural Information Processing Systems*, 37:59716–59754.
- Ziheng Chen, Yue Song, Xiao-Jun Wu, and Nicu Sebe. 2025a. Gyrogroup batch normalization. In *ICLR*.
- Ziheng Chen, Yue Song, Xiaojun Wu, and Nicu Sebe. 2025b. Riemannian transformation layers for general geometries. *OpenReview*.
- Ziheng Chen, Zihan Su, Bernhard Schölkopf, and Nicu Sebe. 2026. Proper velocity neural networks. In *ICLR*.
- Ziheng Chen, Xiao-Jun Wu, Bernhard Schölkopf, and Nicu Sebe. 2025c. Riemannian batch normalization: A gyro approach. *arXiv preprint arXiv:2509.07115*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Taco S Cohen, Mario Geiger, Jonas Kohler, and Max Welling. 2018. Spherical cnns. In *ICLR*.
- Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.
- Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.
- Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *NeurIPS*.
- Gemma Team. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré. 2019. Learning mixed-curvature representations in product spaces. In *ICLR*.
- Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2019. Hyperbolic attention networks. In *ICLR*.
- Yunhui Guo, Xudong Wang, Yubei Chen, and Stella X Yu. 2022. Clipped hyperbolic classifiers are super-hyperbolic classifiers. In *CVPR*.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.
- Isay Katsman, Eric Chen, Sidhanth Holalkere, Anna Asch, Aaron Lou, Ser Nam Lim, and Christopher M De Sa. 2023. Riemannian residual neural networks. In *NeurIPS*.
- Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *NAACL*.
- Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images.
- Shanglin Li, Shiwen Chu, Okan Koç, Yi Ding, Qibin Zhao, Motoaki Kawanabe, and Ziheng Chen. 2026. HEEGNet: Hyperbolic embeddings for EEG. In *ICLR*.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *ACL*.
- Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic graph neural networks. In *NeurIPS*.
- Aaron Lou, Derek Lim, Isay Katsman, Leo Huang, Qingxuan Jiang, Ser Nam Lim, and Christopher M De Sa. 2020. Neural manifold ordinary differential equations. In *NeurIPS*.
- Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163.
- Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*.
- Maximilian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *ICML*.

- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *NAACL*.
- Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.
- Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. 2021. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044.
- Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Ramit Sawhney, Megh Thakkar, Shivam Agarwal, Di Jin, Diyi Yang, and Lucie Flek. 2021. Hypmix: hyperbolic interpolative data augmentation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 9858–9868.
- Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine*, 29(3):93–93.
- Xianglong Shi, Ziheng Chen, Yunhan Jiang, and Nicu Sebe. 2026. Intrinsic lorentz neural network. *arXiv preprint arXiv:2602.23981*.
- Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. 2021. Hyperbolic neural networks++. In *ICLR*.
- Ondrej Skopec, Octavian-Eugen Ganea, and Gary Bécigneul. 2020. Mixed-curvature variational autoencoders. In *ICLR*.
- Mingyang Song, Huafeng Liu, and Liping Jing. 2023. Hyperrank: hyperbolic ranking model for unsupervised keyphrase extraction. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 16070–16080.
- Xingwei Tan, Gabriele Pergola, and Yulan He. 2021. Extracting event temporal relations via hyperbolic geometry. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8065–8077.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Max van Spengler, Erwin Berkhout, and Pascal Mettes. 2023. Poincaré resnet. In *ICCV*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.
- Zekun Wang, Jingjie Zeng, Yingxu Li, Liang Yang, and Hongfei Lin. 2025. Human-inspired obfuscation for model unlearning: Local and global strategies with hyperbolic representations. In *EMNLP (Findings)*, pages 14354–14366.
- Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. 2023. Sgformer: Simplifying and empowering transformers for large-graph representations. In *NeurIPS*.
- Menglin Yang, Ram Samarth B B, Aosong Feng, Bo Xiong, Jiahong Liu, Irwin King, and Rex Ying. 2025. Hyperbolic fine-tuning for large language models. In *NeurIPS*.
- Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. 2024. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *KDD*.
- Yiding Zhang, Xiao Wang, Xunqiang Jiang, Chuan Shi, and Yanfang Ye. 2021. [Hyperbolic graph attention network](#). *IEEE Transactions on Big Data*, 8(6):1690–1701.

Manifold Embedding: A Point-to-Hyperplane Approach

Supplementary Material

Appendix Contents

A	Notation	13
B	Preliminaries	14
B.1	Hyperbolic neural networks	14
B.2	Poincaré ball operations	14
B.3	Sphere model operations	14
B.4	Projected-sphere operations	15
B.5	Lorentz model operations	15
B.6	Poincaré fully connected layers in HNN++	15
C	PHDM formula derivations	16
D	Existence, uniqueness, and injectivity of PHDM	17
E	True geodesic distance preservation	17
F	Implementation and experimental details	18
G	More Experiments	19
H	Use of large language models	20

A Notation

Table 8 summarizes the notation used throughout the paper, including the input variables, geometric models, inner products, and PHDM-specific quantities.

Table 8: Summary of notation used throughout the paper.

Notation	Description
General notation and Euclidean score generation	
d, n	Dimensions of the Euclidean input and manifold output, respectively
$k \in \{1, \dots, n\}$	Coordinate or canonical-hyperplane index
K	Constant sectional curvature, with $K < 0$ for hyperbolic models and $K > 0$ for spherical models
$\text{sgn}(K)$	Sign of the curvature, equal to -1 for $K < 0$ and $+1$ for $K > 0$
$c = K $	Positive curvature magnitude used when the hyperbolic curvature is written as $K = -c$
$\mathbb{R}^n, 0$	n -dimensional Euclidean space and its zero vector
$x \in \mathbb{R}^d$	Euclidean feature supplied to the score layer; elsewhere, x denotes a manifold point when its domain is stated
W, w_k, b, b_k	Euclidean weight matrix, its k -th row, bias vector, and k -th bias
$v(x), v_k(x), v$	Euclidean score vector, its k -th component $v_k(x) = \langle w_k, x \rangle - b_k$, and the resulting signed-distance vector
y, Y	Reconstructed target-manifold point
$e^{(k)}$	k -th standard basis vector
$\langle u, v \rangle, \ u\ _2$	Standard Euclidean inner product and norm, where $\langle u, v \rangle = \sum_i u_i v_i$
g^E, I	Standard Euclidean metric $g^E = \langle \cdot, \cdot \rangle$ and its identity-matrix representation
Riemannian geometry and constant-curvature models	
$\mathcal{M}, T_x \mathcal{M}$	Riemannian manifold and its tangent space at x
g_x^K, λ_x^K	Stereographic metric at x and its conformal factor, with $g_x^K = (\lambda_x^K)^2 g^E$
$d_{\mathcal{M}}(x, y)$	Geodesic distance between x and y on $\mathcal{M}$
$d_{\mathcal{M}}^{\pm}(y, H)$	Signed geodesic distance from y to hyperplane H
$\exp_x(v), \log_x(y)$	Riemannian exponential and logarithmic maps at x
$H_{w,p}^{\mathcal{M}}$	Geodesic hyperplane on $\mathcal{M}$ with normal w and base point p
$H_{e^{(k)},0}^{\mathcal{M}}$	Canonical hyperplane through the origin and orthogonal to $e^{(k)}$
$\mathcal{M}_K^n$	Radius-model family, equal to $\mathbb{L}_K^n$ for $K < 0$ and $\mathbb{S}_K^n$ for $K > 0$
st_K^n	Stereographic-model family, equal to $\mathbb{P}_K^n$ for $K < 0$ and $\mathbb{D}_K^n$ for $K > 0$
$\mathbb{L}_K^n, \mathbb{S}_K^n$	Lorentz hyperboloid and ambient sphere of curvature K
$\mathbb{P}_K^n, \mathbb{D}_K^n$	Poincaré ball and projected-sphere models of curvature K
$x = [x_0, x_s]$	Pole or time coordinate x_0 and spatial component $x_s \in \mathbb{R}^n$ of a radius-model point
$\langle u, v \rangle_{\mathcal{L}}, \ v\ _{\mathcal{L}}$	Lorentz inner product $-u_0 v_0 + \sum_{i=1}^n u_i v_i$ and its induced tangent norm; subscript $\mathcal{L}$ denotes the same quantities in the appendix
$\langle u, v \rangle_K, \ v\ _K$	Curvature-aware inner product and tangent norm for the radius-model family
$\sin_K, \cos_K, \tan_K$	Curvature-aware trigonometric functions, using hyperbolic functions for $K < 0$ and circular functions for $K > 0$
$\arcsin, \text{arsinh}$	Inverse functions of $\sin$ and $\sinh$, respectively. We use the single-valued principal branch $\arcsin : [-1, 1] \rightarrow [-\pi/2, \pi/2]$. In the signed-distance formulas, they convert scaled spatial coordinates into spherical and hyperbolic distances
$x \oplus_K y$	Signed-curvature Mobius addition in a stereographic model
$u = (-x) \oplus_K y$	Mobius displacement from x to y
$\rho_K^{-1}(x)$	Inverse stereographic projection from $\mathbb{D}_K^n$ to $\mathbb{S}_K^n$
PHDM and appendix derivations	
$r_K(v)$	Elementwise inverse-distance vector, using $(-K)^{-1/2} \sinh(\sqrt{-K}v)$ for $K < 0$ and $K^{-1/2} \sin(\sqrt{K}v)$ for $K > 0$
$\text{PHDM}_{\mathcal{M}}(v)$	Parameter-free reconstruction of a point on $\mathcal{M}$ from canonical signed distances v
$s, \alpha, \alpha_{\pm}$	$s = \ r_K(v)\ _2^2$, the stereographic ray coefficient, and the two quadratic roots used in the uniqueness proof
ρ	Positive-curvature feasibility statistic $\ \sin(\sqrt{K}v)\ _2$ used in the stability experiment
p, w, z, a	Hyperplane base point and normal, together with the Euclidean parameters used in the Lorentz reparameterization
A_c, D_c, A_K, D_K	Numerators and denominators of the negative- and positive-curvature Mobius additions
u_p, N_p, D_p, δ_p	Poincaré displacement and intermediate quantities in its signed point-to-hyperplane distance
$q_K, \eta_{\mathbb{S}}, \theta$	Projected-sphere lift denominator, spherical exponential-map scale, and spherical angular distance
$F^c(x; Z, r), w(v)$	HNN++ Poincaré fully connected map with learned parameters (Z, r) and its inverse-distance reconstruction vector

B Preliminaries

After the notation summary, this appendix provides tutorial background on hyperbolic neural networks and collects the constant-curvature operations used in the main text. We write the negative curvature as $K = -c$ with $c > 0$ when following the Poincaré-ball notation of HNN++ (Shimizu et al., 2021). Equivalently, $c = |K|$, and positive-curvature sphere operations use the signed-curvature notation $K > 0$.

B.1 Hyperbolic neural networks

Hyperbolic neural networks (HNNs) are a broad family of models that represent features or perform neural operations on hyperbolic manifolds, rather than a single architecture (Ganea et al., 2018; Peng et al., 2021). Their motivation comes from hyperbolic geometry, where the volume of metric balls grows exponentially with radius, allowing tree-like and hierarchical structures to be embedded with less distortion than in Euclidean space (Nickel and Kiela, 2017, 2018). This property is relevant to NLP because language representations can contain latent hierarchies and asymmetric relations (Tan et al., 2021; Chen et al., 2022). Hyperbolic geometry provides a useful inductive bias when such structures are present.

B.2 Poincaré ball operations

The n -dimensional Poincaré ball of curvature $-c$ is

$$\mathbb{P}_{-c}^n = \{x \in \mathbb{R}^n \mid c\|x\|_2^2 < 1\}, \quad (20)$$

$$\lambda_x^c = \frac{2}{1 - c\|x\|_2^2}.$$

Its metric is conformal to the Euclidean metric, $g_x^c = (\lambda_x^c)^2 I$. The Möbius addition used to move vectors inside the ball is

$$x \oplus_c y = \frac{A_c(x, y)}{D_c(x, y)}, \quad (21)$$

$$A_c(x, y) = (1 + 2c\langle x, y \rangle + c\|y\|_2^2)x + (1 - c\|x\|_2^2)y,$$

$$D_c(x, y) = 1 + 2c\langle x, y \rangle + c^2\|x\|_2^2\|y\|_2^2.$$

With $u = (-x) \oplus_c y$, the geodesic distance is

$$d_{\mathbb{P}_{-c}}(x, y) = \frac{2}{\sqrt{c}} \operatorname{artanh}(\sqrt{c}\|u\|_2). \quad (22)$$

The exponential and logarithmic maps are

$$\exp_x^c(v) = x \oplus_c z_x^c(v), \quad (23)$$

$$z_x^c(v) = \tanh\left(\frac{\sqrt{c}\lambda_x^c\|v\|_2}{2}\right) \frac{v}{\sqrt{c}\|v\|_2},$$

$$\log_x^c(y) = \frac{2}{\sqrt{c}\lambda_x^c} \operatorname{artanh}(\sqrt{c}\|u\|_2) \frac{u}{\|u\|_2}, \quad (24)$$

$$u = (-x) \oplus_c y.$$

Zero-norm cases are interpreted by continuity.

A Poincaré hyperplane with base point $p \in \mathbb{P}_{-c}^n$ and normal $a \in T_p\mathbb{P}_{-c}^n$ can be written as (Ganea et al., 2018; Shimizu et al., 2021)

$$\tilde{H}_{a,p}^c = \{x \in \mathbb{P}_{-c}^n \mid \langle a, (-p) \oplus_c x \rangle = 0\}. \quad (25)$$

The corresponding signed point-to-hyperplane distance is

$$u_p(x) = (-p) \oplus_c x,$$

$$\delta_p(x) = \frac{N_p(x)}{D_p(x)}, \quad (26)$$

$$N_p(x) = 2\sqrt{c}\langle a, u_p(x) \rangle,$$

$$D_p(x) = (1 - c\|u_p(x)\|_2^2)\|a\|_2,$$

$$d_{\mathbb{P}_{-c}}^\pm(x, \tilde{H}_{a,p}^c) = \frac{1}{\sqrt{c}} \operatorname{arsinh}(\delta_p(x)).$$

For the output hyperplane through the origin and orthogonal to the k -th axis, $p = 0$ and $a = e_k$, this reduces to

$$h_k(y) = \frac{2\sqrt{c}y_k}{1 - c\|y\|_2^2}, \quad (27)$$

$$d_{\mathbb{P}_{-c}}^\pm(y, \bar{H}_k) = \frac{1}{\sqrt{c}} \operatorname{arsinh}(h_k(y)),$$

$$\bar{H}_k = \{y \in \mathbb{P}_{-c}^n \mid y_k = 0\}.$$

B.3 Sphere model operations

For positive curvature $K > 0$, the ambient sphere model is

$$\mathbb{S}_K^n = \left\{x \in \mathbb{R}^{n+1} \mid \langle x, x \rangle = \frac{1}{K}\right\}, \quad (28)$$

$$T_x\mathbb{S}_K^n = \{v \mid \langle v, x \rangle = 0\}.$$

For $x, y \in \mathbb{S}_K^n$ and $v \in T_x\mathbb{S}_K^n$, with $\|v\|_2 = \sqrt{\langle v, v \rangle}$, the distance is

$$d_{\mathbb{S}_K}(x, y) = \frac{1}{\sqrt{K}} \arccos(K\langle x, y \rangle). \quad (29)$$

The exponential and principal logarithmic maps are

$$\eta_{\mathbb{S}} = \sqrt{K}\|v\|_2, \quad (30)$$

$$\exp_x^{\mathbb{S}_K}(v) = \cos(\eta_{\mathbb{S}})x + \frac{\sin(\eta_{\mathbb{S}})}{\eta_{\mathbb{S}}}v,$$

$$\log_x^{\mathbb{S}_K}(y) = \frac{\theta}{\sin \theta} (y - K\langle x, y \rangle x), \quad (31)$$

$$\theta = \arccos(K\langle x, y \rangle).$$

The logarithmic map is interpreted on the principal branch away from the antipodal point. The canonical spherical hyperplane associated with the k -th spatial axis is

$$\bar{H}_k^{\mathbb{S}} = \{y \in \mathbb{S}_K^n \mid y_k = 0\}. \quad (32)$$

Its signed point-to-hyperplane distance is

$$d_{\mathbb{S}_K}^{\pm}(y, \bar{H}_k^{\mathbb{S}}) = \frac{1}{\sqrt{K}} \arcsin(\sqrt{K}y_k). \quad (33)$$

B.4 Projected-sphere operations

For positive curvature $K > 0$, the projected-sphere model uses a stereographic parameterization

$$\begin{aligned} \mathbb{D}_K^n &= \mathbb{R}^n, \\ \lambda_x^K &= \frac{2}{1 + K\|x\|_2^2}. \end{aligned} \quad (34)$$

The point $x \in \mathbb{D}_K^n$ corresponds to the ambient sphere point

$$\begin{aligned} q_K(x) &= 1 + K\|x\|_2^2, \\ \rho_K^{-1}(x) &= \left(\frac{1 - K\|x\|_2^2}{\sqrt{K}q_K(x)}, \frac{2x}{q_K(x)} \right). \end{aligned} \quad (35)$$

Its metric is conformal to the Euclidean metric, $g_x^K = (\lambda_x^K)^2 I$. The positive-curvature Möbius addition is (Skopek et al., 2020)

$$\begin{aligned} x \oplus_K y &= \frac{A_K(x, y)}{D_K(x, y)}, \\ A_K(x, y) &= (1 - 2K\langle x, y \rangle - K\|y\|_2^2)x \\ &\quad + (1 + K\|x\|_2^2)y, \\ D_K(x, y) &= 1 - 2K\langle x, y \rangle + K^2\|x\|_2^2\|y\|_2^2. \end{aligned} \quad (36)$$

With $u = (-x) \oplus_K y$, the geodesic distance is

$$d_{\mathbb{D}_K}(x, y) = \frac{2}{\sqrt{K}} \arctan(\sqrt{K}\|u\|_2). \quad (37)$$

The exponential and logarithmic maps are

$$\begin{aligned} \exp_x^{\mathbb{D}_K}(v) &= x \oplus_K z_x^K(v), \\ z_x^K(v) &= \tan\left(\frac{\sqrt{K}\lambda_x^K\|v\|_2}{2}\right) \frac{v}{\sqrt{K}\|v\|_2}, \end{aligned} \quad (38)$$

$$\begin{aligned} \log_x^{\mathbb{D}_K}(y) &= \frac{2}{\sqrt{K}\lambda_x^K} \arctan\left(\sqrt{K}\|u\|_2\right) \frac{u}{\|u\|_2}, \\ u &= (-x) \oplus_K y. \end{aligned} \quad (39)$$

Zero-norm cases are interpreted by continuity. For the canonical projected-sphere hyperplane

$$\bar{H}_k^{\mathbb{D}} = \{y \in \mathbb{D}_K^n \mid y_k = 0\}, \quad (40)$$

the signed point-to-hyperplane distance used by PHDM is

$$d_{\mathbb{D}_K}^{\pm}(y, \bar{H}_k^{\mathbb{D}}) = \frac{1}{\sqrt{K}} \arcsin\left(\frac{2\sqrt{K}y_k}{1 + K\|y\|_2^2}\right). \quad (41)$$

B.5 Lorentz model operations

Let $\langle x, y \rangle_L = -x_0y_0 + \sum_{k=1}^n x_ky_k$. The Lorentz model of curvature $-c$ is

$$\begin{aligned} \mathbb{L}_{-c}^n &= \{x \in \mathbb{R}^{n+1} \mid \ell_c(x) = -1/c, x_0 > 0\}, \\ \ell_c(x) &= \langle x, x \rangle_L, \\ T_x\mathbb{L}_{-c}^n &= \{v \mid \langle x, v \rangle_L = 0\}. \end{aligned} \quad (42)$$

For $x, y \in \mathbb{L}_{-c}^n$, the distance is

$$d_{\mathbb{L}_{-c}}(x, y) = \frac{1}{\sqrt{c}} \operatorname{arcosh}(-c\langle x, y \rangle_L). \quad (43)$$

For $v \in T_x\mathbb{L}_{-c}^n$ with $\|v\|_L = \sqrt{\langle v, v \rangle_L}$, the exponential and logarithmic maps are

$$\begin{aligned} \eta_L &= \sqrt{c}\|v\|_L, \\ \exp_x^L(v) &= \cosh(\eta_L)x + \frac{\sinh(\eta_L)}{\eta_L}v, \end{aligned} \quad (44)$$

$$\begin{aligned} \log_x^L(y) &= \frac{\theta}{\sinh \theta} (y + c\langle x, y \rangle_L x), \\ \theta &= \operatorname{arcosh}(-c\langle x, y \rangle_L). \end{aligned} \quad (45)$$

The canonical Lorentz hyperplane associated with the k -th spatial axis is

$$\bar{H}_k^L = \{y \in \mathbb{L}_{-c}^n \mid y_k = 0\}, \quad (46)$$

and its signed point-to-hyperplane distance is

$$d_{\mathbb{L}_{-c}}^{\pm}(y, \bar{H}_k^L) = \frac{1}{\sqrt{c}} \operatorname{arsinh}(\sqrt{c}y_k). \quad (47)$$

B.6 Poincaré fully connected layers in HNN++

HNN++ reformulates a Euclidean fully connected layer by interpreting each output scalar as a signed distance to a hyperplane and then replacing the Euclidean output hyperplanes with Poincaré hyperplanes (Shimizu et al., 2021). Given an input $x \in \mathbb{P}_{-c}^n$, HNN++ first computes generalized scalar

scores $v_k(x)$ using its Poincaré MLR parameterization. The Poincaré FC layer then sets

$$w(v) = (c^{-1/2} \sinh(\sqrt{c}v_1(x)), \dots, c^{-1/2} \sinh(\sqrt{c}v_n(x))). \quad (48)$$

It returns

$$F^c(x; Z, r) = \frac{w(v)}{1 + \sqrt{1 + c\|w(v)\|_2^2}}. \quad (49)$$

By Eq. (27), the signed distance from $F^c(x; Z, r)$ to the k -th Poincaré output hyperplane through the origin equals $v_k(x)$. This is the same inverse-distance reconstruction used by PHDM for the Poincaré model, with $K = -c$ and $r_K(v) = c^{-1/2} \sinh(\sqrt{c}v)$. HNN++ contrasts this construction with the tangent-space alternative $\exp_0^c(A \log_0^c(x)) \oplus_c b$: after a nonzero Mobius bias, the resulting level sets are generally not Poincaré hyperplanes.

C PHDM formula derivations

This section gives the algebra behind the four closed-form PHDM maps in the main text. The construction starts from the signed-distance constraints

$$d_{\mathcal{M}}^{\pm}(y, H_{e^{(k)},0}^{\mathcal{M}}) = v_k, \quad k = 1, \dots, n, \quad (50)$$

and solves them for the target point y .

Radius models. For the Lorentz model with $K < 0$, the signed distance is

$$d_{\mathbb{L}_K}^{\pm}(y, H_{e^{(k)},0}^{\mathbb{L}_K}) = \frac{1}{\sqrt{-K}} \operatorname{arsinh}(\sqrt{-K}y_k). \quad (51)$$

Setting this equal to v_k and applying $\sinh$ to both sides gives

$$y_k = (-K)^{-1/2} \sinh(\sqrt{-K}v_k) = r_{K,k}. \quad (52)$$

Writing $y = [y_0, r_K(v)]$ and using the hyperboloid constraint $\langle y, y \rangle_{\mathcal{L}} = 1/K$ gives

$$-y_0^2 + \|r_K(v)\|_2^2 = \frac{1}{K}. \quad (53)$$

The future-sheet condition selects the positive root, yielding

$$y_0 = \sqrt{-\frac{1}{K} + \|r_K(v)\|_2^2}. \quad (54)$$

This proves the Lorentz PHDM formula.

For the sphere with $K > 0$, the principal signed distance is

$$d_{\mathbb{S}_K}^{\pm}(y, H_{e^{(k)},0}^{\mathbb{S}_K}) = \frac{1}{\sqrt{K}} \arcsin(\sqrt{K}y_k). \quad (55)$$

On the principal branch, the same inversion gives

$$y_k = K^{-1/2} \sin(\sqrt{K}v_k) = r_{K,k}. \quad (56)$$

Writing $y = [y_0, r_K(v)]$ and using $\|y\|_2^2 = 1/K$ gives

$$y_0 = \sqrt{\frac{1}{K} - \|r_K(v)\|_2^2}. \quad (57)$$

The north-hemisphere branch chooses the positive root. Exact reconstruction requires

$$\left| \sqrt{K}v_k \right| \leq \frac{\pi}{2}, \quad \sum_{k=1}^n \sin^2(\sqrt{K}v_k) \leq 1. \quad (58)$$

The first condition makes the elementwise inverse single-valued, and the second condition makes the remaining sphere entry real.

Stereographic models. For the Poincaré model with $K < 0$, the canonical signed distance has the form

$$\begin{aligned} d_{\mathbb{P}_K}^{\pm}(y, H_{e^{(k)},0}^{\mathbb{P}_K}) &= \frac{1}{\sqrt{-K}} \operatorname{arsinh}\left(\frac{2\sqrt{-K}y_k}{1 + K\|y\|_2^2}\right). \end{aligned} \quad (59)$$

Solving the signed-distance equation gives

$$\begin{aligned} \frac{2y_k}{1 + K\|y\|_2^2} &= (-K)^{-1/2} \sinh(\sqrt{-K}v_k) \\ &= r_{K,k}. \end{aligned} \quad (60)$$

In vector form,

$$\frac{2y}{1 + K\|y\|_2^2} = r_K(v). \quad (61)$$

The solution lies on the ray spanned by $r_K(v)$, so write $y = \alpha r_K(v)$. With $s = \|r_K(v)\|_2^2$, the scalar equation becomes

$$Ks\alpha^2 - 2\alpha + 1 = 0. \quad (62)$$

The branch satisfying $y = 0$ when $r_K(v) = 0$ is

$$\alpha = \frac{1}{1 + \sqrt{1 - Ks}}. \quad (63)$$

Therefore,

$$y = \frac{r_K(v)}{1 + \sqrt{1 - K\|r_K(v)\|_2^2}}. \quad (64)$$

For $K < 0$, the quantity under the square root is larger than one for finite v , so the construction is globally defined up to floating-point overflow in the sinh computation. For the projected sphere with $K > 0$, the principal signed distance is

$$d_{\mathbb{D}_K}^\pm(y, H_{e^{(k)},0}^{\mathbb{D}_K}) = \frac{1}{\sqrt{K}} \arcsin\left(\frac{2\sqrt{K}y_k}{1 + K\|y\|_2^2}\right). \quad (65)$$

The same algebra gives

$$\begin{aligned} \frac{2y}{1 + K\|y\|_2^2} &= r_K(v), \\ r_{K,k} &= K^{-1/2} \sin(\sqrt{K}v_k). \end{aligned} \quad (66)$$

Solving $y = \alpha r_K(v)$ again yields

$$y = \frac{r_K(v)}{1 + \sqrt{1 - K\|r_K(v)\|_2^2}}. \quad (67)$$

For $K > 0$, this is the principal branch and requires $1 - K\|r_K(v)\|_2^2 \geq 0$, which is equivalent to the joint feasibility condition above.

D Existence, uniqueness, and injectivity of PHDM

The preceding derivations construct a point whose signed distances to the canonical hyperplanes equal v . We now show that this point is the unique solution on each stated branch.

Radius models. For the Lorentz model, substituting the constructed spatial coordinates into the signed-distance formula gives

$$\frac{1}{\sqrt{-K}} \operatorname{arsinh}\left(\sinh\left(\sqrt{-K}v_k\right)\right) = v_k. \quad (68)$$

Thus the constructed point satisfies every signed-distance equation, which proves existence. Since arsinh is strictly increasing, each spatial coordinate y_k is uniquely determined, and the future-sheet constraint uniquely fixes the positive value of y_0 . For the sphere, Eq. (58) keeps the elementwise inverse on the principal branch and makes the remaining coordinate real. Substitution again proves existence, while the strict monotonicity of $\arcsin$ on its principal branch uniquely fixes each y_k and the north-hemisphere constraint uniquely fixes the nonnegative value of y_0 .

Stereographic models. For both the Poincaré and projected-sphere models, direct inversion of all signed-distance equations gives

$$\frac{2y}{1 + K\|y\|_2^2} = r_K(v). \quad (69)$$

The denominator is positive on both model domains. Hence, if $r_K(v) = 0$, then $y = 0$. Otherwise, every solution is a positive scalar multiple of $r_K(v)$, so the ray property follows directly from the signed-distance equations rather than being assumed. Substituting the ray form and $s = \|r_K(v)\|_2^2$ used in the preceding derivation gives

$$\begin{aligned} Ks\alpha^2 - 2\alpha + 1 &= 0, \\ \alpha_- &= \frac{1}{1 + \sqrt{1 - Ks}}, \\ \alpha_+ &= \frac{1}{1 - \sqrt{1 - Ks}}. \end{aligned} \quad (70)$$

For $K < 0$, α_+ is negative, so α_- is the only root compatible with the positive scalar multiple above. For $K > 0$ with $K\|r_K(v)\|_2^2 < 1$, both roots are positive, but they represent the north and south sphere branches, and the principal north branch selects α_- . At the boundary $K\|r_K(v)\|_2^2 = 1$, the roots coincide at 1, giving the unique equator point. This proves existence and uniqueness for both stereographic models on their stated branches.

Injectivity. Two points on the same stated branch with identical signed distances have the same uniquely determined spatial coordinates in the radius models. In the stereographic models, they have the same $r_K(v)$ and the same unique admissible branch root, so they also coincide. Therefore, the canonical signed distances determine at most one point on each stated branch. The substitutions above also show that applying the signed-distance equations to $\text{PHDM}(v)$ returns v throughout the feasible score domain. Conversely, reconstructing a point from its canonical signed distances returns that point on the stated branch. Hence PHDM and the canonical signed-distance operation are mutually inverse bijections between the feasible score domain and each stated branch.

E True geodesic distance preservation

True geodesic distance. Let $H_{a,p}^{\mathcal{M}}$ be a geodesic hyperplane with base point p and normal a . RMLR defines the intrinsic point-to-hyperplane distance as follows (Chen et al., 2024, Eq. (6)):

$$d_{\mathcal{M}}(y, H_{a,p}^{\mathcal{M}}) = \inf_{z \in H_{a,p}^{\mathcal{M}}} d_{\mathcal{M}}(y, z). \quad (71)$$

On a branch where the orientation induced by a is well defined, its signed counterpart is

$$d_{\mathcal{M}}^{\pm}(y, H_{a,p}^{\mathcal{M}}) = \text{sgn}(\langle \log_p(y), a \rangle_p) \cdot d_{\mathcal{M}}(y, H_{a,p}^{\mathcal{M}}). \quad (72)$$

Computing this infimum can be difficult because it generally requires a non-convex optimization over a curved manifold. RMLR therefore starts from the following Euclidean identity (Chen et al., 2024, Eq. (7)):

$$d_{\mathbb{E}}(y, H_{a,p}) = \sin(\angle ypz^{\star}) d_{\mathbb{E}}(y, p), \quad (73)$$

where $z^{\star}$ is the Euclidean projection of y onto the hyperplane. RMLR then extends this construction through the logarithmic map, yielding (Chen et al., 2024, Eq. (10))

$$\tilde{d}_{\mathcal{M}}(y, H_{a,p}^{\mathcal{M}}) = \frac{|\langle \log_p(y), a \rangle_p|}{\|a\|_p}. \quad (74)$$

This quantity is exactly the distance from $\log_p(y)$ to the corresponding linear hyperplane in $T_p\mathcal{M}$. For signed output coordinates, we use its oriented counterpart

$$\tilde{d}_{\mathcal{M}}^{\pm}(y, H_{a,p}^{\mathcal{M}}) = \frac{\langle \log_p(y), a \rangle_p}{\|a\|_p}, \quad (75)$$

whose absolute value equals $\tilde{d}_{\mathcal{M}}$. However, the logarithmic map is not generally an isometry between arbitrary pairs of manifold points, so this tangent-space pseudo-distance need not attain the intrinsic infimum above. For example, on the unit sphere, let $u = (\bar{e}^{(1)} + \bar{e}^{(2)})/2$ for orthonormal tangent vectors $\bar{e}^{(1)}$ and $\bar{e}^{(2)}$. For $Y = \text{Exp}_o(u)$, the first signed pseudo-distance is $1/2$, whereas the corresponding true signed distance is $\arcsin(\sin(1/\sqrt{2})/\sqrt{2}) \neq 1/2$.

ExpMap reconstructs pseudo-distances. Let $\{\bar{e}^{(k)}\}_{k=1}^n$ be the canonical tangent basis at o normalized with respect to g_o , so that $\langle \bar{e}^{(j)}, \bar{e}^{(k)} \rangle_o = \delta_{jk}$. For the stereographic models, $\lambda_o^K = 2$, and hence $\bar{e}^{(k)} = e^{(k)}/2$ when $e^{(k)}$ denotes the Euclidean standard basis. This normalization does not change the associated hyperplane. Riemannian Transformation Layers use signed pseudo-distances as the output-coordinate constraints of their fully connected layer (Chen et al., 2025b, Definition 4.1). If

$$u = \sum_{k=1}^n v_k \bar{e}^{(k)} \quad (76)$$

lies in a normal neighborhood of o on which $\log_o$ and Exp_o are inverse, their canonical-basis solution is (Chen et al., 2025b, Theorem 4.2)

$$Y = \text{Exp}_o(u). \quad (77)$$

It then satisfies

$$\tilde{d}_{\mathcal{M}}^{\pm}(Y, H_{\bar{e}^{(k)},o}^{\mathcal{M}}) = \langle \log_o(Y), \bar{e}^{(k)} \rangle_o = v_k. \quad (78)$$

Accordingly, the Euclidean-to-manifold embedding can be written as (Chen et al., 2025b, Proposition 7.1)

$$Y = \text{Exp}_o(u(x)), \quad u(x) = \sum_{k=1}^n (Ax + b)_k \bar{e}^{(k)}. \quad (79)$$

For the negative-curvature models considered here, the origin exponential map is globally injective. For the positive-curvature models, the reconstruction statement requires $\|u(x)\|_o < \pi/\sqrt{K}$, which excludes the spherical cut locus. Thus, ExpMap reconstructs a point whose signed tangent-space pseudo-distances, rather than necessarily its true signed geodesic distances, equal the input scores.

PHDM reconstructs true distances. PHDM instead derives each canonical point-to-hyperplane distance by directly minimizing the manifold two-point distance over the hyperplane, and then inverts these intrinsic signed distances. On every stated feasible branch, the preceding results give

$$\sum_{k=1}^n \left(d_{\mathcal{M}}^{\pm}(\text{PHDM}_{\mathcal{M}}(v), H_{e^{(k)},0}^{\mathcal{M}}) - v_k \right)^2 = 0. \quad (80)$$

Thus, the origin exponential map is the pseudo-distance-based special case of the same reconstruction framework, whereas PHDM denotes the true-distance construction. This identity guarantees exact reconstruction in the canonical true-distance coordinates on the stated feasible branches.

F Implementation and experimental details

Controlled replacement. Across all comparative experiments, PHDM changes only the Euclidean-to-manifold input map. The upstream encoder, downstream manifold layers, loss, optimizer, scheduler, data preprocessing, and trainable parameter budget are kept identical to the corresponding baseline row. Thus any trainable parameters remain in

the surrounding model, while PHDM itself adds no learned normal, offset, or bias.

PHDM LoRA implementation. For the arithmetic-reasoning experiment, PHDM LoRA follows the LoRA/HypLoRA adapter placement and parameter budgets used by the corresponding reported rows. A standard LoRA adapter adds a low-rank Euclidean residual update to a frozen linear transformation (Hu et al., 2022). HypLoRA inserts a hyperbolic adapter by mapping the low-rank Euclidean adapter feature into the hyperbolic space, applying the hyperbolic transformation, and mapping the result back to the Euclidean residual stream (Yang et al., 2025). PHDM LoRA keeps that surrounding adapter structure, curvature setting, trainable low-rank matrices, and Euclidean output residual path fixed, but replaces the Euclidean-to-hyperbolic input map inside the adapter with the negative-curvature PHDM map. Concretely, for the adapter feature z produced by the low-rank LoRA branch, PHDM LoRA interprets z_k as the signed distance to the k -th canonical hyperplane, applies the PHDM construction with the same hyperbolic curvature as the matched HypLoRA row. Therefore the “PHDM LoRA” rows test only the geometric input-map substitution and do not add learned PHDM parameters, additional hyperplane normals, or extra adapter rank.

Numerical stability benchmark. The stability experiment uses FP32 arithmetic, signed curvature $K = -1$ or $K = +1$, and 4096 samples in dimension 16. For negative curvature, vectors are sampled by random directions and controlled radius. We report the relative recovery error at radius 10 and the failure rate at radius 50. For positive curvature, samples are drawn inside the feasible signed-distance domain by controlling

$$\rho = \|\sin(\sqrt{K}v)\|_2, \quad (81)$$

with $\rho = 0.99$ in the table. This probes the boundary of the exact PHDM domain rather than testing arbitrary tangent vectors outside the domain.

Positive-curvature feasibility. For sphere and projected-sphere PHDM, exact signed-distance preservation requires both the principal component-wise constraint and the joint constraint

$$\left|\sqrt{K}v_k\right| \leq \frac{\pi}{2}, \quad \sum_{k=1}^n \sin^2(\sqrt{K}v_k) \leq 1. \quad (82)$$

When an implementation receives infeasible positive-curvature inputs, clipping or radial scaling

can restore manifold validity. These safeguards are useful engineering choices, but they no longer preserve the signed-distance equations exactly for the modified components.

Comparative experiments. Image-classification experiments follow the ResNet-18 HCNN setting on CIFAR-10, CIFAR-100, and Tiny-ImageNet, replacing the original input map in the Hybrid Poincaré and Hybrid Lorentz variants. Graph node-classification experiments follow Hypformer and replace only its Lorentz input map. Cora link-prediction experiments follow the RResNet and RResNet Graph settings and evaluate all four target geometries. LLM arithmetic experiments use MAWPS, SVAMP, GSM8K, and AQUA (Koncel-Kedziorski et al., 2016; Patel et al., 2021; Cobbe et al., 2021; Ling et al., 2017) and report PHDM LoRA under the same parameter budgets as the corresponding LoRA and HypLoRA rows. Unless stated otherwise, results are reported as means and standard deviations over five runs, and all training experiments are conducted on A100 64G GPUs.

Runtime and visualization. The runtime comparison uses the same Lorentz image-classification replacement as the accuracy experiment and reports average per-epoch wall-clock time. The embedding visualization uses the same held-out CIFAR-100 test features for PHDM and ExpMap0, applies a shared PCA-UMAP visualization basis, and computes the class-separation scores in the original high-dimensional feature space rather than in the two-dimensional UMAP display.

G More Experiments

Additional Hypformer geometries. Table 9 extends the controlled Hypformer replacement to Poincaré, spherical, and projected-sphere input maps. PHDM improves over the matched ExpMap baseline in all 12 additional settings. The Poincaré ExpMap baseline remains collapsed on Airport in the expanded five-run evaluation, with an F1-score of 55.34 ± 8.31 , whereas PHDM reaches 97.33 ± 0.40 . Across the other 11 settings, the gains range from +1.00 to +2.46 points.

Curvature sensitivity. Table 10 evaluates PHDM over curvature magnitudes $|K| \in \{0.5, 1, 2, 4, 10\}$ while keeping the remaining Cora link-prediction settings. Performance varies by less than one ROC-AUC point for every geometry, with spans of 0.85, 0.70, 0.48, and 0.83 for

Table 9: Additional Hypformer input-map replacements beyond the Lorentz setting in the main text. Airport reports F1, while Cora, Citeseer, and PubMed report accuracy, all in percent. Results are means and standard deviations over five runs. Δ is PHDM minus ExpMap.

Manifold	ExpMap	PHDM	Δ
Airport (F1)			
Poincaré	55.34 $\pm$ 8.31	97.33$\pm$0.40	+41.99
Sphere	96.29 $\pm$ 0.28	97.86$\pm$0.16	+1.57
Projected sphere	95.29 $\pm$ 0.21	97.75$\pm$0.16	+2.46
Cora (accuracy)			
Poincaré	82.48 $\pm$ 1.47	84.32$\pm$1.21	+1.84
Sphere	83.56 $\pm$ 1.72	84.64$\pm$1.58	+1.08
Projected sphere	83.68 $\pm$ 1.80	84.96$\pm$1.32	+1.28
Citeseer (accuracy)			
Poincaré	68.64 $\pm$ 0.85	70.08$\pm$0.46	+1.44
Sphere	69.28 $\pm$ 0.84	70.28$\pm$0.50	+1.00
Projected sphere	69.60 $\pm$ 0.55	70.64$\pm$0.54	+1.04
PubMed (accuracy)			
Poincaré	78.52 $\pm$ 2.18	79.60$\pm$1.92	+1.08
Sphere	78.32 $\pm$ 2.55	79.32$\pm$2.41	+1.00
Projected sphere	78.20 $\pm$ 2.37	79.24$\pm$2.35	+1.04

Table 10: Curvature sensitivity of PHDM on Cora link prediction. Results are ROC AUC in percent, reported as mean and standard deviation over five runs. Span is the difference between the maximum and minimum mean across the evaluated curvature magnitudes.

	Lorentz	Poincaré	Sphere	Projected sphere
0.5	91.33 $\pm$ 0.31	91.28 $\pm$ 0.60	93.34 $\pm$ 0.12	90.53 $\pm$ 1.01
1	91.67 $\pm$ 0.15	91.42 $\pm$ 0.45	93.09 $\pm$ 0.44	89.70 $\pm$ 0.52
2	91.08 $\pm$ 0.39	91.69 $\pm$ 0.43	92.86 $\pm$ 0.24	90.43 $\pm$ 0.77
4	90.82 $\pm$ 0.71	91.51 $\pm$ 0.33	92.88 $\pm$ 0.65	90.47 $\pm$ 0.23
10	91.27 $\pm$ 0.39	90.99 $\pm$ 0.75	93.16 $\pm$ 0.30	89.99 $\pm$ 0.35
Span	0.85	0.70	0.48	0.83

Lorentz, Poincaré, sphere, and projected sphere, respectively. The sphere has the smallest span, while all four geometries remain stable throughout the evaluated range.

Table 11: Fine-grained computational overhead and throughput in the LLM setting on an A100 64G GPU. Throughput is measured in tokens per second, and peak memory is reported for training and inference in GiB.

Metric	LoRA	HypLoRA	PHDM LoRA
	54,132,736	54,133,016	54,132,876
Trainable parameters	0.7058%	0.7058%	0.7058%
Train step (ms)	717.7	784.6	777.4
Train throughput (tok/s)	2853.5	2451.6	2634.3
Prefill throughput (tok/s)	8886.1	8867.1	8927.0
Generation throughput (tok/s)	32.60	21.25	21.80
Peak memory, train/infer (GiB)	43.89/14.49	45.07/14.48	44.34/14.49

LLM efficiency. Table 11 reports fine-grained training and inference efficiency under essentially identical trainable parameter budgets. Compared with HypLoRA, PHDM LoRA reduces the train-step time from 784.6 to 777.4 ms and improves training, prefill, and generation throughput by

7.5%, 0.7%, and 2.6%, respectively. It also reduces peak training memory by 0.73 GiB, while peak inference memory remains effectively unchanged at 14.49 GiB. These results show that replacing HypLoRA’s input map with PHDM reduces its computational overhead without materially changing the parameter budget.

H Use of large language models

We use large language models to aid or polish writing.