Hyperbolic Busemann Neural Networks

Ziheng Chen[1,2], Bernhard Schölkopf[2] & Nicu Sebe[1]
[1] University of Trento, [2] MPI-IS

Abstract

Hyperbolic spaces provide a natural geometry for representing hierarchical and tree-structured data due to their exponential volume growth. To leverage these benefits, neural networks require intrinsic and efficient components that operate directly in hyperbolic space. In this work, we lift two core components of neural networks, Multinomial Logistic Regression (MLR) and Fully Connected (FC) layers, into hyperbolic space via Busemann functions, resulting in Busemann MLR (BMLR) and Busemann FC (BFC) layers with a unified mathematical interpretation. BMLR provides compact parameters, a point-to-horosphere distance interpretation, batch-efficient computation, and a Euclidean limit, while BFC generalizes FC and activation layers with comparable complexity. Experiments on image classification, genome sequence learning, node classification, and link prediction demonstrate improvements in effectiveness and efficiency over prior hyperbolic layers. The code is available at `https://github.com/GitZH-Chen/HBNN`.

1. Introduction

Hyperbolic representations have recently delivered strong performance across different applications because the exponential volume growth of negatively curved manifolds enables low-distortion embeddings of tree-like and hierarchical structure [46]. These advantages have been validated in computer vision [3, 4, 20, 24, 25, 31, 34, 42, 54, 56, 57], graph learning [2, 8, 22, 53], multimodal learning [18, 48], recommendation systems [60], astronomy [10], genome sequence learning [33], natural language processing [23, 27, 30, 46, 47, 59], and brain signal decoding [40]. These empirical successes highlight the need for principled and efficient hyperbolic neural network components that fully leverage hyperbolic geometry.

Among canonical formulations, the Poincaré ball and the Lorentz model are the most widely used, owing to their closed-form Riemannian operators. To support hyperbolic deep learning, several key building blocks in neural networks have been generalized to Poincaré or Lorentz spaces, including attention [11, 27, 58], batch normalization [3, 13, 15, 43], linear feed-forward layers [11, 23, 51], activation [3, 23], residual blocks [31, 32, 56], Multinomial Logistics Regression (MLR) [3, 45, 51], and graph convolution [2, 8, 16, 41]. Among these components, MLR classification and Fully Connected (FC) layers play a fundamental role in final decision-making and feature transformation.

Recently, hyperplanes and point-to-hyperplane distances [38] have been adopted to construct hyperbolic MLR in both Poincaré [23, 51] and Lorentz [3] models. Ganea et al. [23, Sec. 3.1] introduced the first Poincaré MLR based on Poincaré hyperplanes, but the formulation suffers from over-parameterization and lacks batch efficiency. Shimizu et al. [51, Sec. 3.1] alleviated these issues through a reparameterization strategy. Building on these ideas, Bdeir et al. [3, Sec. 4.3] proposed a Lorentz MLR. However, its hyperplane is defined by the ambient Minkowski space, which is model-specific and may distort Lorentzian geometry.

For hyperbolic FC layers, three main formulations exist. Ganea et al. [23, Sec. 3.2] introduced Möbius matrix-vector multiplication through the tangent space on the Poincaré ball. Shimizu et al. [51, Sec. 3.2] further proposed the Poincaré FC layer, defined intrinsically but restricted to the Poincaré model. On the Lorentz model, Chen et al. [11, Sec. 2.2] constructed a Lorentz FC layer by applying linear transformations in the ambient Minkowski space followed by projection onto the Lorentz model. Thus, Möbius and Lorentz FC rely on flat-space (tangent or ambient) approximations that could distort intrinsic geometry, whereas Poincaré FC is intrinsic but model-specific.

On the other hand, the Busemann function and its level sets, horospheres, have emerged as powerful intrinsic tools for hyperbolic learning. They enjoy convenient metric properties [6, Ch. II.8] and admit closed-form expressions on both the Poincaré and Lorentz models [5, Prop. 9]. These operators have supported several hyperbolic algorithms, including SVM [21], PCA [9], Sliced Wasserstein distances [5], and prototype learning [26]. We also note that Nguyen et al. [45, Cor. 4.3] proposed a Poincaré MLR based on the Busemann function. However, its induced point-to-hyperplane distance is pseudo, coincides with the true distance only in Euclidean geometry, remains over-parameterized, and is not batch efficient.

These observations motivate intrinsic and batch-efficient formulations for MLR and FC layers that can operate on both the Poincaré ball and the Lorentz model. To address this need, we propose *Busemann Multinomial Logistics Regression (BMLR)* and *Busemann Fully Connected (BFC)* layers, two Busemann-based components for hyperbolic networks. Our contributions are summarized as follows:

- We introduce BMLR, deriving intrinsic logits directly from Busemann functions with a point-to-horosphere distance interpretation. BMLR uses a compact per-class parameterization, eliminates manifold-valued parameters in prior MLRs, remains batch-efficient, and recovers Euclidean MLR as curvature tends to zero.
- We develop BFC layers by generalizing the FC and activation layers through the Busemann function, providing intrinsic constructions on both the Poincaré and Lorentz models. BFC preserves comparable complexity and parameter counts, and recovers Euclidean FC in the zero curvature limit.
- We provide empirical validation across image classification, genome sequence learning, node classification, and link prediction. BMLR and BFC generally outperform existing hyperbolic layers. BMLR shows particularly large gains as the number of classes increases, and the Lorentz BMLR is the fastest among all hyperbolic MLRs.

2. Preliminaries

Hyperbolic geometry. An n-dimensional hyperbolic space is a Riemannian manifold with constant negative sectional curvature $K < 0$. There exist five models of hyperbolic space [7, Sec. 7]. Among them, the Poincaré ball and Lorentz models are the most widely used. The *Poincaré ball model* [39, p. 62] is $\mathbb{P}_K^n = \{x \in \mathbb{R}^n : \|x\|^2 < -1/K\}$ with Riemannian metric $g_x(u,v) = (\lambda_x^K)^2 \langle u, v \rangle$, where $\lambda_x^K = \frac{2}{1+K\|x\|^2}$. Here $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the Euclidean inner product and norm. The origin is the zero vector $\mathbf{0} \in \mathbb{P}_K^n$. The *Lorentz model* [39, p. 62], also called the hyperboloid model, is $\mathbb{L}_K^n = \{x \in \mathbb{R}^{n+1} : \langle x, x \rangle_{\mathcal{L}} = 1/K,\ x_t > 0\}$, where $\langle x, y \rangle_{\mathcal{L}} = -x_t y_t + \langle x_s, y_s \rangle$ is the Lorentzian inner product and $\|x\|_{\mathcal{L}} = \sqrt{\langle x, x \rangle_{\mathcal{L}}}$ is the induced Lorentzian norm. Following the convention of special relativity [49, §3.1], we write $x = [x_t, x_s^\top]^\top$ with temporal component $x_t \in \mathbb{R}$ and spatial component $x_s \in \mathbb{R}^n$. The canonical origin is $\overline{\mathbf{0}} = [\sqrt{1/|K|}, 0, \ldots, 0]^\top \in \mathbb{L}_K^n$. Throughout, we write $\mathcal{H}_K^n \in \{\mathbb{P}_K^n, \mathbb{L}_K^n\}$ for a hyperbolic space.

Geodesic, geodesic ray, and asymptote. *Geodesics* are length-minimizing curves. We consider unit speed geodesics. A *geodesic ray* is a geodesic defined for $t \in [0, \infty)$. Two geodesic rays are *asymptotic* if the distance between corresponding points remains bounded as t tends to infinity [6, II. Def. 8.1]. This notion generalizes Euclidean parallel lines in two respects: (i) Euclidean parallels have constant separation, whereas asymptotic rays have bounded separation; (ii) Euclidean parallels never intersect, and asymptotic rays do not intersect as well [6, II. Prop. 8.2].

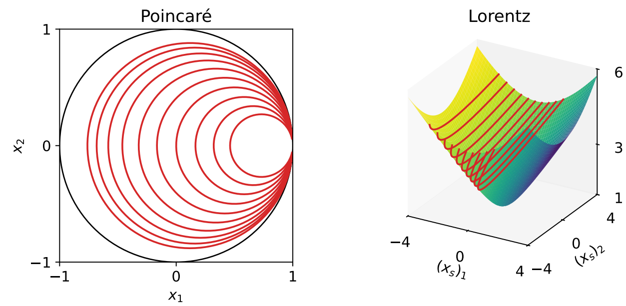

Figure 1. Illustration: red curves are different horospheres of B^v.

Busemann function. Let γ be a geodesic ray and let $\mathrm{d}(\cdot, \cdot)$ denote the geodesic distance. The *Busemann function* [6, II. Def. 8.17] associated with γ is defined as

$$\forall x \in \mathcal{H}_K^n, \quad B^\gamma(x) = \lim_{t \to \infty} \left(\mathrm{d}\left(x, \gamma(t)\right) - t \right). \tag{1}$$

In hyperbolic spaces, this limit exists [6, II. Lem. 8.18]. In Euclidean space, the Busemann function associated with the geodesic $\gamma(t) = tv$ that starts at $\mathbf{0}$ with unit direction $v \in \mathbb{S}^{n-1} = \{v \in \mathbb{R}^n \mid \|v\| = 1\}$ is

$$\forall x \in \mathbb{R}^n, \quad B^v(x) = -\langle x, v \rangle, \tag{2}$$

which coincides, up to a sign, with the inner product. Hence, the Busemann function provides an intrinsic generalization of the inner product to manifolds. Up to an additive constant, it is equivalent among asymptotic geodesic rays [6, II. Cor. 8.20], independent of the starting point. Consequently, we write $B^v(x)$ for the ray that emanates from the origin $e \in \mathcal{H}_K^n$ with unit velocity $v \in \mathbb{S}^{n-1} \subset T_e\mathcal{H}_K^n$. For $v \in \mathbb{S}^{n-1}$ and $x \in \mathcal{H}_K^n$, closed forms of the Poincaré and Lorentz Busemann functions [5, Prop. 9] are

$$\mathbb{P}_K^n : \ B^v(x) = \frac{1}{\sqrt{-K}} \log \left(\frac{\left\| v - \sqrt{-K} x \right\|^2}{1 + K \|x\|^2} \right), \tag{3}$$

$$\mathbb{L}_K^n : \ B^v(x) = \frac{1}{\sqrt{-K}} \log \left(\sqrt{-K} \left(x_t - \langle x_s, v \rangle \right) \right). \tag{4}$$

Horosphere. The level sets of a Busemann function are called *horospheres*, the hyperbolic counterpart of Euclidean hyperplanes. In Euclidean space, for a unit direction v, the hyperplanes $H_\tau^v = \{x \in \mathbb{R}^n \mid \langle x, v \rangle = \tau\}$ with $\tau \in \mathbb{R}$ are parallel. Analogously, for fixed v, the horospheres $H_\tau^v = \{x \in \mathcal{H}_K^n \mid B^v(x) = \tau\}$ are equidistant (see Thm. 3.3). Fig. 1 illustrates such horospheres, and Tab. 1 summarizes the correspondence of the above concepts between Euclidean and hyperbolic geometries.

Riemannian & gyrovector operators. The hyperbolic space $\mathcal{H}_K^n$ provides closed-form expressions for standard

Table 1. Generalization: Euclidean vs. hyperbolic.

Euclidean	Hyperbolic
Straight line	Geodesic
Parallel lines	Asymptotic geodesic rays
Inner product $\langle v, x\rangle$	Busemann function $-B^v(x)$
Hyperplane	Horosphere
Parallel hyperplanes with $v \in \mathbb{S}^{n-1}$	Horospheres with $v \in \mathbb{S}^{n-1} \cap T_e\mathcal{H}_K^n$

Riemannian operators. It also supports a gyrovector structure that extends vector space operations to hyperbolic geometry. On the Poincaré ball $\mathbb{P}_K^n$, this structure corresponds to the Möbius gyrovector space [55, Ch. 6.14], denoted as $\{\mathbb{P}_K^n, \oplus_{\mathrm{M}}, \odot_{\mathrm{M}}\}$. As shown by Chen et al. [15, Props. 24 and 25], the Lorentz model also admits a closed-form gyrovector space, denoted as $\{\mathbb{L}_K^n, \oplus_{\mathbb{L}}, \odot_{\mathbb{L}}\}$. Please refer to Sec. B.4 for detailed expressions.

3. Busemann multinomial logistic regression

We begin by reformulating the Euclidean Multinomial Logistics Regression (MLR), then lift it to hyperbolic space via the Busemann function, introducing *Busemann MLR (BMLR)*. We also present a point-to-hyperplane interpretation. Finally, we compare BMLR with existing hyperbolic MLRs, highlighting our advantages in geometric fidelity, parameterization, and computational efficiency.

3.1. Formulation

The Euclidean MLR $\operatorname{softmax}(Ax+b)$ computes the multinomial probability for each class $k \in \{1, \ldots, C\}$ given an input $x \in \mathbb{R}^n$. It admits the inner product form:

$$\forall k,\ p(y=k \mid x) = \frac{\exp(\langle a_k, x\rangle + b_k)}{\sum_{j=1}^{C} \exp(\langle a_j, x\rangle + b_j)}, \tag{5}$$

where $a_k \in \mathbb{R}^n$ and $b_k \in \mathbb{R}$ are the weight and bias for the class k. We write $p(y=k \mid x) \propto \exp(u_k(x))$ with $u_k(x) = \langle a_k, x\rangle + b_k$. Decomposing the weight vector into a magnitude $\alpha_k = \|a_k\| > 0$ and a unit direction $v_k = \frac{a_k}{\|a_k\|} \in \mathbb{S}^{n-1}$, each logit is

$$u_k(x) = \alpha_k \langle v_k, x\rangle + b_k. \tag{6}$$

As shown in Tab. 1, the Busemann function naturally generalizes the Euclidean inner product. Analogously to Eq. (6), we define the hyperbolic logits via the Busemann function, yielding BMLR:

$$\forall k,\ p(y=k \mid x) = \frac{\exp(u_k(x))}{\sum_{j=1}^{C} \exp(u_j(x))}, \tag{7}$$

$$u_k(x) = -\alpha_k B^{v_k}(x) + b_k, \tag{8}$$

with $\alpha_k > 0$, $v_k \in \mathbb{S}^{n-1}$, and $b_k \in \mathbb{R}$ as parameters.

The following results show that, as $K \to 0^-$, both the Poincaré and Lorentz BMLRs reduce to the Euclidean MLR.

Theorem 3.1 (Limits of BMLRs)**.** *As $K \to 0^-$, the hyperbolic Busemann functions converge to the Euclidean inner product:*

$$\text{(Poincaré)} \quad B^v(x) \xrightarrow{K\to 0^-} -2\langle v, x\rangle, \tag{9}$$

$$\text{(Lorentz)} \quad B^v(x) \xrightarrow{K\to 0^-} -\langle v, x_s\rangle. \tag{10}$$

The hyperbolic BMLRs converge to the Euclidean MLR:

$$\text{(Poincaré)} \quad u_k(x) \xrightarrow{K\to 0^-} 2\alpha_k \langle v_k, x\rangle + b_k, \tag{11}$$

$$\text{(Lorentz)} \quad u_k(x) \xrightarrow{K\to 0^-} \alpha_k \langle v_k, x_s\rangle + b_k, \tag{12}$$

Proof. The proof is provided in Sec. F.1. □

Remark 3.2 (Intuition). On the Poincaré ball, letting $K \to 0^-$ recovers Euclidean geometry [52, App. A.4.2]. For the Lorentz model, as $K \to 0^-$, the temporal coordinate diverges while the spatial component approaches $\mathbb{R}^n$, making $\mathbb{L}_K^n$ converge to a Euclidean space. Consistently, the Poincaré and Lorentz Busemann functions and the associated BMLR logits reduce to their Euclidean counterparts, providing a natural generalization of Euclidean MLR.

3.2. Geometric interpretation

As shown by Lebanon and Lafferty [38, Sec. 5], the Euclidean MLR can be reformulated by point-to-hyperplane distances:

$$u_k(x) = \operatorname{sign}(\langle a_k, x\rangle + b_k)\, \|a_k\|\, \mathrm{d}(x, H_{a_k,b_k}), \tag{13}$$

with $a_k \in \mathbb{R}^n$ and $b_k \in \mathbb{R}$. Here, $H_{a_k,b_k} = \{x \in \mathbb{R}^n : \langle a_k, x\rangle + b_k = 0\}$ denotes the margin hyperplane, and the point-to-hyperplane distance is $\mathrm{d}(x, H_{a_k,b_k}) = \frac{|\langle a_k, x\rangle + b_k|}{\|a_k\|}$. We next show that our BMLR also admits such an interpretation through point-to-horosphere distances.

We first introduce these distances in Hadamard spaces, metric spaces with favorable properties (see Def. B.8), where Euclidean and hyperbolic geometries arise as special cases.

Theorem 3.3 (Hadamard horosphere distance)**.** *Let $(\mathcal{X}, \mathrm{d})$ be a Hadamard space, and $B^\gamma : \mathcal{X} \to \mathbb{R}$ be the Busemann function associated with a geodesic ray $\gamma : [0, \infty) \to \mathcal{X}$. For any $\tau_1, \tau_2 \in \mathbb{R}$, define the horospheres by*

$$H^\gamma_{\tau_i} = \{x \in \mathcal{X} \mid B^\gamma(x) = \tau_i\},\ i = 1, 2. \tag{14}$$

The distance between these horospheres is constant:

$$\mathrm{d}\left(H^\gamma_{\tau_1}, H^\gamma_{\tau_2}\right) = \mathrm{d}\left(H^\gamma_{\tau_2}, H^\gamma_{\tau_1}\right) = |\tau_2 - \tau_1|. \tag{15}$$

In particular, the point-to-horosphere distance is

$$\mathrm{d}(x, H^\gamma_\tau) = |B^\gamma(x) - \tau|,\ \forall x \in \mathcal{X}. \tag{16}$$

Table 2. Comparison of C-class MLR. In Dist, **Real** means the point-to-hyperplane distance is the real distance, obtained by $\inf_{y \in H} \mathrm{d}(x, y)$, where H is a hyperplane and d is the geodesic distance; **Pseudo** denotes a surrogate that coincides with the real distance only in Euclidean geometry. Compact params indicate whether each logit avoids an additional manifold-valued parameter. Batch efficiency indicates whether the MLR can avoid inefficient per-class loops in implementation (see App. C). In #Params, we highlight the heaviest in **red**. In FLOPs, we mark the slowest in **red** and the fastest in **green**.

Method	Logit $u_k(x),\ \forall k \in \{1, \ldots, C\}$	Space	Dist	#Params	Compact params	FLOPs	Batch efficiency
Euclidean MLR	$\langle a_k, x \rangle + b_k$, with $a_k \in \mathbb{R}^n, b_k \in \mathbb{R}$	$\mathbb{R}^n$	**Real**	$C(n+1)$	✓	$C(2n)$	✓
Poincaré MLR [23, Eq. (25)]	$\frac{\lambda_{p_k}^K \lVert a_k \rVert}{\sqrt{-K}} \sinh^{-1}\left(\frac{2\sqrt{-K} \langle -p_k \oplus_{\mathrm{M}} x, a_k \rangle}{\left(1 + K \lVert -p_k \oplus_{\mathrm{M}} x \rVert^2\right) \lVert a_k \rVert} \right)$, with $p_k \in \mathbb{P}_K^n, a_k \in T_{p_k}\mathbb{P}_K^n$	$\mathbb{P}_K^n$	**Real**	$C(2n)$	✗	$C(19n+29)$	✗
Poincaré MLR [51, Eq. (6)]	$\frac{2}{\sqrt{-K}} \alpha_k \sinh^{-1}(\alpha - \beta)$, $\alpha = \lambda_x^K \sqrt{-K} \langle x, v_k \rangle \cosh\left(2\sqrt{-K} b_k\right)$, $\beta = \left(\lambda_x^K - 1\right) \sinh\left(2\sqrt{-K} b_k\right)$, with $\alpha_k > 0, v_k \in \mathbb{S}^{n-1}, b_k \in \mathbb{R}$	$\mathbb{P}_K^n$	**Real**	$C(n+2)$	✓	$C(4n+52)$	✓
Pseudo-Busemann MLR [45, Cor. 4.3]	$-d(x, p_k) \frac{B^{v_k}(-p_k \oplus_{\mathrm{M}} x)}{\lVert -p_k \oplus_{\mathrm{M}} x \rVert}$, with $p_k \in \mathbb{P}_K^n, v_k \in \mathbb{S}^{n-1}$	$\mathbb{P}_K^n$	**Pseudo**	$C(2n)$	✗	$C(19n+34)$	✗
Lorentz MLR [3, Eq. (12)]	$\frac{1}{\sqrt{-K}} \operatorname{sign}(\alpha) \beta \left\lvert \sinh^{-1}\left(\sqrt{-K} \frac{\alpha}{\beta} \right) \right\rvert$, $\alpha = \cosh\left(\sqrt{-K} b_k\right) \langle z_k, x_s \rangle - \sinh\left(\sqrt{-K} b_k\right)$, $\beta = \sqrt{\lVert \cosh(\sqrt{-K} b_k) z_k \rVert^2 - (\sinh(\sqrt{-K} b_k) \lVert z_k \rVert)^2}$, with $z_k \in \mathbb{R}^n,\ b_k \in \mathbb{R}$	$\mathbb{L}_K^n$	**Real**	$C(n+1)$	✓	$C(4n+52)$	✓
BMLR	$-\alpha_k B^{v_k}(x) + b_k$, with $\alpha_k > 0, v_k \in \mathbb{S}^{n-1}, b_k \in \mathbb{R}$	$\mathbb{P}_K^n$ $\mathbb{L}_K^n$	**Real**	$C(n+2)$	✓	$\mathbb{P}_K^n : C(6n+12)$ $\mathbb{L}_K^n : C(2n+12)$	✓

Proof. A Hadamard space does not need to be a manifold. It extends manifolds with nonpositive curvature to the broader setting of metric spaces. A brief review of metric geometry and the complete proof are given in Secs. B.2 and F.2. □

Corollary 3.4 (Point-to-horosphere distance)**.** *In a hyperbolic space $\mathcal{H}_K^n \in \{\mathbb{P}_K^n, \mathbb{L}_K^n\}$ with nonpositive curvature $K \leq 0$, the point-to-horosphere distance is*

$$\mathrm{d}\left(x, H_\tau^v\right) = \left| B^v(x) - \tau \right|, \tag{17}$$

where $H_\tau^v = \{x \mid B^v(x) = \tau\}$ denotes the horosphere with respect to the direction $v \in \mathbb{S}^{n-1}$.

A Euclidean hyperplane $H_{a,b}$ can be parameterized as $H_{v,\alpha,b} = \{x \in \mathbb{R}^n : \alpha \langle v, x \rangle + b = 0\}$ with $v \in \mathbb{S}^{n-1}$, $\alpha > 0$, and $b \in \mathbb{R}$. Similarly, we parameterize a hyperbolic horosphere as

$$H_{v,\alpha,b} = \left\{x \in \mathcal{H}_K^n \mid -\alpha B^v(x) + b = 0\right\}, \tag{18}$$

with $v \in \mathbb{S}^{n-1}$, $\alpha > 0$, and $b \in \mathbb{R}$. With this parameterization, we extend Eq. (13) to hyperbolic spaces:

$$u_k(x) = \operatorname{sign}_k \alpha_k \, \mathrm{d}\left(x, H_{v_k,\alpha_k,b_k}\right) \tag{19}$$

where $\operatorname{sign}_k = \operatorname{sign}\left(-\alpha_k B^{v_k}(x) + b_k\right)$, and $\{v_k \in \mathbb{S}^{n-1}, \alpha_k > 0, b_k \in \mathbb{R}\}$ are parameters for class k. By Cor. 3.4, the point-to-horosphere distance is

$$\mathrm{d}\left(x, H_{v,\alpha,b}\right) = \frac{\left|-\alpha B^v(x) + b\right|}{\alpha}. \tag{20}$$

By Eq. (20), Eq. (19) equals the exact BMLR logit in Eq. (8).

Remark 3.5 (Generality). Since $B^v(x) = -\langle v, x \rangle$ in Euclidean geometry, Eqs. (18) to (20) naturally generalize to their Euclidean counterparts. We also acknowledge Fan et al. [21, Eq. (2) and Prop. 3.1], who used horospheres and point-to-horosphere distances to construct a hyperbolic SVM. However, they considered only the unit Poincaré ball with the specific curvature $K = -1$, which is a special case of our Eqs. (18) and (20).

3.3. Comparison

Based on the point-to-hyperplane reformulation in Eq. (13), recent work extended MLR to the Poincaré [23, 45, 51] and Lorentz [3] models. Ganea et al. [23, Sec. 3.1] introduced the first Poincaré MLR by replacing the Euclidean point-to-hyperplane distance with its hyperbolic counterpart, where the hyperplane is defined by geodesics and the resulting distance is the real point-to-hyperplane distance, obtained as an infimum over the hyperplane. However, the formulation is not batch efficient (see App. C). It also requires per-class parameters $a_k \in T_{p_k}\mathbb{P}_K^n$ and $p_k \in \mathbb{P}_K^n$, which leads to over-parameterization. Shimizu et al. [51, Sec. 3.1] alleviated such issues via re-parameterization. Bdeir et al. [3, Sec. 4.3] further developed a Lorentz MLR, but its hyperplanes are defined by the ambient Minkowski space, which is tailored to the Lorentz model and does not fully respect the intrinsic hyperbolic geometry. Moreover, Nguyen et al. [45, Cor. 4.3] proposed a Poincaré MLR based on the Busemann function. We refer to it as Pseudo-Busemann MLR, as the induced point-to-hyperplane distance is pseudo, coinciding with the real point-to-hyperplane distance only in Euclidean geometry. It also suffers from over-parameterization and is not batch efficient.

As summarized in Tab. 2[1], BMLR unifies advantages that prior hyperbolic MLRs offer only partially. In particular, BMLR respects the real point-to-horosphere distance, uses compact parameters without an additional manifold-valued point, attains the lowest FLOPs on $\mathbb{L}_K^n$ and a competitive cost on $\mathbb{P}_K^n$, and supports batch-efficient computation. On $\mathbb{L}_K^n$, its FLOPs are even close to those of the Euclidean MLR.

4. Busemann fully connected layer

We first revisit the Euclidean Fully Connected (FC) layer from a geometric perspective, then propose the hyperbolic *Busemann FC (BFC)* layer with its manifestations in the Poincaré and Lorentz models.

4.1. Formulation

Euclidean FC layers. An FC affine transformation $\mathcal{F}: \mathbb{R}^n \ni x \mapsto y = Ax + b \in \mathbb{R}^m$ can be expressed element-wise as $y_k = \langle a_k, x \rangle + b_k$ with $a_k \in \mathbb{R}^n$ and $b_k \in \mathbb{R}$. As shown by Shimizu et al. [51, Sec. 3.2] and Chen et al. [14, Sec. 3.1], the LHS y_k can be interpreted as the signed distance from y to the hyperplane passing through the origin and orthogonal to the k-th axis of the output space. Hence, the FC layer can be expressed as

$$\bar{\mathrm{d}}\left(y, H_{e_k,0}\right) = \langle a_k, x \rangle + b_k, \ \forall 1 \leq k \leq m, \tag{21}$$

where $\bar{\mathrm{d}}\left(y, H_{e_k,0}\right) = \operatorname{sign}\left(\langle e_k, y \rangle\right) \mathrm{d}\left(y, H_{e_k,0}\right)$ is the signed point-to-hyperplane distance, and $H_{e_k,0} = \{y \in \mathbb{R}^m \mid \langle e_k, y \rangle = 0\}$ is a hyperplane with $e_k \in \mathbb{R}^m$ as the vector whose k-th element is 1 and all others are 0.

Lifting to hyperbolic space. To extend Eq. (21) into hyperbolic space, the RHS can be readily replaced by Eq. (8), as it generalizes the term $\langle a_k, x \rangle + b_k$. For the LHS, a natural idea is to use the signed point-to-horosphere distance. However, as detailed in App. D, this may fail to admit a solution for y. We therefore follow the point-to-hyperplane distance in [23, Thm. 5] for the Poincaré model and the one in [3, Eq. (44)] for the Lorentz model. Given $x \in \mathcal{H}_K^n$, the hyperbolic BFC layer $\mathcal{F}: \mathcal{H}_K^n \ni x \mapsto y \in \mathcal{H}_K^m$ is given by solving y via the following m equations:

$$\bar{\mathrm{d}}\left(y, H_{e_k,e}\right) = u_k(x), \ \forall 1 \leq k \leq m, \tag{22}$$

where $u_k(x) = -\alpha_k B^{v_k}(x) + b_k$ with $\{\alpha_k > 0, v_k \in \mathbb{S}^{n-1}, b_k \in \mathbb{R}\}$ as parameters. Here, $\bar{\mathrm{d}}\left(y, H_{e_k,e}\right)$ is the hyperbolic signed distance from y to the hyperplane passing through the origin $e \in \mathcal{H}_K^n$. Next, we show that the above implicit definition has an explicit solution for the output y.

[1]Relative to [45, Def. 4.2, Cor. 4.3, and App. B.1.2], the Pseudo-Busemann MLR written here includes an additional sign $-$; this is intentional and matches their official implementation.

Theorem 4.1. *Given an input $x \in \mathbb{P}_K^n$, the Poincaré BFC layer $\mathcal{F}: \mathbb{P}_K^n \to \mathbb{P}_K^m$ is given by*

$$y = \frac{\omega}{1 + \sqrt{1 - K \|\omega\|^2}}, \ \omega = \left[\frac{\sinh\left(\sqrt{-K} u_k(x)\right)}{\sqrt{-K}}\right]_{k=1}^m,$$

where $u_k(x) = -\alpha_k B^{v_k}(x) + b_k$ with $\{\alpha_k > 0, v_k \in \mathbb{S}^{n-1}, b_k \in \mathbb{R}\}$ as parameters for $\{k = 1, \ldots, m\}$.

Proof. The proof, together with $H_{e_k,e}$ and $\bar{\mathrm{d}}\left(y, H_{e_k,e}\right)$, is provided in Sec. F.3, which is inspired by Shimizu et al. [51, App. D.3]. □

Theorem 4.2. *Given an input $x \in \mathbb{L}_K^n$, the Lorentz BFC layer $\mathcal{F}: \mathbb{L}_K^n \to \mathbb{L}_K^m$ is given by*

$$y = \begin{bmatrix} y_t \\ y_s \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{1}{-K} + \|y_s\|^2} \\ \frac{1}{\sqrt{-K}} \sinh\left(\sqrt{-K} u(x)\right) \end{bmatrix}$$

where $u(x) = \left(u_1(x), \ldots, u_m(x)\right)^\top$ with $u_k(x) = -\alpha_k B^{v_k}(x) + b_k$. Here, $\{\alpha_k > 0, v_k \in \mathbb{S}^{n-1}, b_k \in \mathbb{R}\}$ are parameters for $\{k = 1, \ldots, m\}$.

Proof. The proof, along with the hyperplane and point-to-hyperplane distance, is provided in Sec. F.4. □

Analogously to Thm. 3.1, our BFC layers converge to their Euclidean counterparts as $K \to 0^-$.

Theorem 4.3 (Limits of BFC layers). *As $K \to 0^-$, the hyperbolic BFC layer $\mathcal{H}_K^n \ni x \mapsto y \in \mathcal{H}_K^m$ reduces to a Euclidean FC layer:*

$$\text{(Poincaré)} \quad y_k \xrightarrow{K \to 0^-} \alpha_k \langle v_k, x \rangle + \frac{1}{2} b_k, \tag{23}$$

$$\text{(Lorentz)} \quad (y_s)_k \xrightarrow{K \to 0^-} \alpha_k \langle v_k, x_s \rangle + b_k. \tag{24}$$

Proof. The proof is provided in Sec. F.5. □

4.2. Generalization

When incorporating an activation function $\phi: \mathbb{R} \to \mathbb{R}$, we can jointly express the Euclidean FC and activation layers, which yields Eq. (21) with $u_k(x) = \phi\left(\langle a_k, x \rangle + b_k\right)$. Accordingly, we extend the hyperbolic BFC by inserting the activation into Eq. (22):

$$\bar{\mathrm{d}}\left(y, H_{e_k,e}\right) = \phi\left(u_k(x)\right), \ \forall 1 \leq k \leq m. \tag{25}$$

This is reflected in Thms. 4.2 and 4.3 by replacing every $u_k(x)$ with $\phi\left(-\alpha_k B^{v_k}(x) + b_k\right)$. Moreover, inspired by

Table 3. Comparison of hyperbolic FC layers. For simplicity, BFC layers do not involve the gyroaddition and assume ϕ is the identity map, which is in line with the Möbius and Lorentz FC layers.

Method	$\mathcal{F}: \mathcal{H}_K^n \ni x \mapsto y \in \mathcal{H}_K^m$	Space	Methodology	Parameters	#Params	FLOPs
Möbius [23, Eq. 27]	$\frac{1}{\sqrt{-K}} \tanh\left(\frac{\lVert Wx \rVert}{\lVert x \rVert} \tanh^{-1}\left(\sqrt{-K}\,\lVert x \rVert\right)\right) \frac{Wx}{\lVert Wx \rVert}$	$\mathbb{P}_K^n$	Tangent	$W \in \mathbb{R}^{m \times n}$	mn	$2nm + 2n$ $+2m + 24$
Poincaré FC [51, Eq. (7)]	$y = \frac{\omega}{1 + \sqrt{1 - K \lVert \omega \rVert^2}},\ \omega_k = \frac{\sinh\left(\sqrt{-K} u_k(x)\right)}{\sqrt{-K}}$, with $u_k(x)$ in Tab. 2	$\mathbb{P}_K^n$	Poincaré geometry	$\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1}$, $b_k \in \mathbb{R}$, for $k = 1, \ldots, m$	$m(n+2)$	$4nm + 71m + 4$
Lorentz FC [11, Eq. (3)]	$y = \begin{bmatrix} \sqrt{\lVert \psi(Wx, v) \rVert^2 - 1/K} \\ \psi(Wx, v) \end{bmatrix}$, $\psi(Wx, v) = \lambda\sigma\left(v^\top x + b'\right) \frac{W\phi(x) + b}{\lVert W\phi(x) + b \rVert}$ with ϕ and σ as the activation and sigmoid	$\mathbb{L}_K^n$	Ambient Minkowski	$W \in \mathbb{R}^{m \times (n+1)}$, $v \in \mathbb{R}^{n+1}, b \in \mathbb{R}^m$, $b' \in \mathbb{R},\ \lambda > 0$	$m(n+1) + m$ $+(n+1) + 2$	$2nm + 8m$ $+2n + 10$
BFC	$y = \frac{\omega}{1 + \sqrt{1 - K \lVert \omega \rVert^2}},\ \omega = \frac{\sinh\left(\sqrt{-K} u(x)\right)}{\sqrt{-K}}$; $y_s = \frac{1}{\sqrt{-K}} \sinh\left(\sqrt{-K} u(x)\right),\ y_t = \sqrt{\frac{1}{-K} + \lVert y_s \rVert^2}$, with $u_k(x) = \phi(-\alpha_k B^{v_k}(x) + b_k)$	$\mathbb{P}_K^n$ $\mathbb{L}_K^n$	Busemann	$\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1}$, $b_k \in \mathbb{R}$, for $k = 1, \ldots, m$	$m(n+2)$	$6nm + 29m + 4$ $2nm + 30m + 2$

the Poincaré Möbius transformation [23, Sec. 3.2], a BFC transformation could be further followed by a gyroaddition $\oplus_{\mathcal{H}}$: $\mathcal{H}_K^n \ni x \mapsto \mathcal{F}(x) \oplus_{\mathcal{H}} b \in \mathcal{H}_K^m$ with $b \in \mathcal{H}_K^m$ as a gyro bias. For example, the Lorentz BFC layer is generalized as

$$\mathbb{L}_K^n \ni x \mapsto y = \begin{bmatrix} \sqrt{\frac{1}{-K} + \lVert y_s \rVert^2} \\ \frac{1}{\sqrt{-K}} \sinh\left(\sqrt{-K} u(x)\right) \end{bmatrix} \oplus_{\mathbb{L}} b \in \mathbb{L}_K^m,$$

where $u_k(x) = \phi\left(-\alpha_k B^{v_k}(x) + b_k\right)$ with parameters $\{\alpha_k > 0, v_k \in \mathbb{S}^{n-1}, b_k \in \mathbb{R}\}_{k=1}^m$ and $b \in \mathbb{L}_K^m$.

4.3. Comparison

Tab. 3 compares BFC with prior hyperbolic FC layers. BFC faithfully respects hyperbolic geometry, whereas the Möbius and Lorentz FC layers apply Euclidean transformations in the tangent or ambient Minkowski space, which can distort intrinsic geometry. BFC also offers flexibility across models, while Poincaré FC and Lorentz FC are tailored to their respective models. In addition, BFC uses a comparable parameterization and maintains $\mathcal{O}(nm)$ FLOPs. On $\mathbb{L}_K^n$, its FLOPs are $\mathcal{O}(2mn)$, matching the fastest layers.

5. Experiments

We first compare BMLRs with prior hyperbolic MLRs on three architectures: ResNet-18 (image classification), CNN (genome sequences), and HGCN (node classification). We then compare BFC with prior hyperbolic FC layers on link prediction. All experiments use both the Poincaré and Lorentz models.

5.1. Image classification

Setup. Following Bdeir et al. [3, Sec. 5.1] and Guo et al. [28, Sec. 4], we use a hybrid architecture with a ResNet-18 [29] backbone and an MLR head. We compare Euclidean MLR with hyperbolic variants in both models. In Poincaré, we evaluate Poincaré MLR (PMLR) re-parameterized by Shimizu et al. [51], Pseudo Busemann MLR (PBMLR-P) [45], and our BMLR-P. In Lorentz, we evaluate Lorentz MLR (LMLR) [3] and our BMLR-L. For hyperbolic MLRs, we map the ResNet-18 features to the target hyperbolic space before classification. We evaluate on CIFAR-10 [36], CIFAR-100 [36], Tiny-ImageNet [37], and ImageNet-1k [17]. On the first three datasets, we conduct five-fold experiments. More details appear in Sec. E.1.

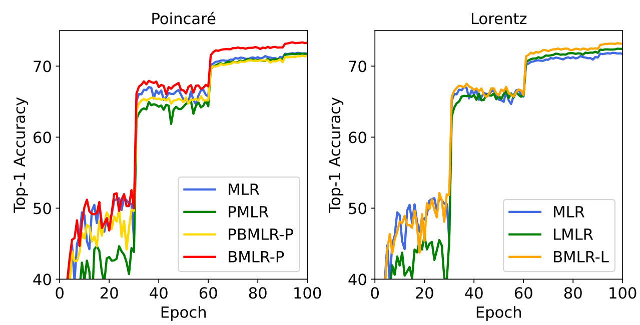

Figure 2. Validation accuracy curves on ImageNet-1k.

Results. Tab. 4 reports top-1 validation accuracy, fit time per epoch, and classifier-head parameters. Fig. 2 presents the ImageNet-1k accuracy curves. Overall, BMLR-P and BMLR-L consistently outperform prior hyperbolic MLRs with comparable parameters. Within each hyperbolic model, the accuracy margin over prior hyperbolic MLRs increases with the number of classes, from CIFAR-10 to CIFAR-100 and Tiny-ImageNet, with the largest gains on ImageNet-1k. This demonstrates the advantage of BMLR as task complexity increases. Besides, PBMLR-P uses approximately double the head parameters and is markedly slower due to complex batch-inefficient computation (see App. C), whereas BMLR-L achieves the fastest fit time among all hyperbolic MLRs.

Table 4. Top-1 image classification accuracy (%) of MLR methods on the ResNet-18 backbone. Best results within each hyperbolic model are in **bold**. Fit time (s/epoch) highlights: fastest hyperbolic MLR in **green**, slowest MLR in **red**. #Params denotes the number of parameters in the MLR head, with the largest marked in **red**.

Space	Method	CIFAR-10 (Num. classes: 10)			CIFAR-100 (Num. classes: 100)			Tiny-ImageNet (Num. classes: 200)			ImageNet-1k (Num. classes: 1000)		
		Acc	Fit Time	#Params	Acc	Fit Time	#Params	Acc	Fit Time	#Params	Acc	Fit Time	#Params
$\mathbb{R}^n$	MLR	95.14 ± 0.12	10.66	5.13K	77.72 ± 0.15	10.60	51.30K	65.19 ± 0.12	69.17	102.60K	71.87	2263.12	513K
$\mathbb{P}^n_K$	PMLR	95.04 ± 0.13	11.94	5.14K	77.19 ± 0.50	12.11	51.40K	64.93 ± 0.38	71.90	102.80K	71.77	2300.11	514K
	PBMLR-P	95.23 ± 0.08	**21.92**	**10.24K**	77.78 ± 0.15	**76.84**	**102.40K**	65.43 ± 0.27	**336.58**	**204.80K**	71.46	**3907.12**	**1024K**
	BMLR-P	**95.32 ± 0.14**	12.01	5.14K	**78.10 ± 0.35**	12.13	51.40K	**66.16 ± 0.19**	71.98	102.80K	**73.36**	2300.77	514K
$\mathbb{L}^n_K$	LMLR	94.98 ± 0.12	11.55	5.13K	78.03 ± 0.21	11.72	51.30K	65.63 ± 0.10	69.27	102.60K	72.46	2277.17	513K
	BMLR-L	**95.25 ± 0.02**	**11.08**	5.14K	**78.07 ± 0.26**	**11.22**	51.40K	**65.99 ± 0.14**	**69.19**	102.80K	**73.24**	**2276.53**	514K

Table 5. Genomic MCC of MLR methods under the CNN backbone. Best results within each hyperbolic model are in **bold**.

Benchmark	Task	Dataset	Num. classes	$\mathbb{P}^n_K$ PMLR	$\mathbb{P}^n_K$ PBMLR-P	$\mathbb{P}^n_K$ BMLR	$\mathbb{L}^n_K$ LMLR	$\mathbb{L}^n_K$ BMLR-L
TEB	Retrotransposons	LTR Copia	2	75.34 ± 1.02	74.37 ± 1.48	**76.73 ± 1.08**	73.01 ± 1.07	**75.86 ± 1.52**
		LINEs	2	85.54 ± 0.61	85.92 ± 0.65	**86.05 ± 1.08**	83.14 ± 0.80	**86.72 ± 0.58**
		SINEs	2	95.30 ± 0.85	95.34 ± 1.58	**95.99 ± 0.74**	**96.70 ± 0.87**	96.29 ± 0.59
	DNA transposons	CMC-EnSpm	2	83.39 ± 0.56	83.62 ± 1.00	**84.03 ± 0.71**	81.78 ± 1.05	**84.15 ± 1.00**
		hAT-Ac	2	89.38 ± 0.90	**89.86 ± 0.54**	89.62 ± 0.74	88.94 ± 0.69	**90.70 ± 0.51**
	Pseudogenes	processed	2	72.45 ± 1.49	71.99 ± 2.04	**73.09 ± 1.66**	**73.71 ± 1.76**	73.32 ± 1.65
		unprocessed	2	75.37 ± 2.27	71.99 ± 1.47	**75.71 ± 1.89**	74.54 ± 1.98	**76.15 ± 1.61**
GUE	Core Promoter Detection	tata	2	**80.95 ± 1.47**	79.32 ± 2.44	80.29 ± 1.63	80.90 ± 1.15	**81.76 ± 1.16**
		notata	2	70.02 ± 0.52	**70.60 ± 0.75**	70.48 ± 0.35	**71.26 ± 0.56**	70.43 ± 0.39
		all	2	67.64 ± 0.77	68.02 ± 0.63	**68.50 ± 0.61**	67.63 ± 0.56	**68.36 ± 1.07**
	Promoter Detection	tata	2	80.30 ± 1.59	80.27 ± 2.71	**82.83 ± 1.69**	**83.27 ± 1.95**	82.55 ± 1.54
		notata	2	92.63 ± 0.36	**93.05 ± 0.32**	92.75 ± 0.51	91.74 ± 0.57	**92.60 ± 0.49**
		all	2	90.53 ± 0.50	**90.79 ± 0.77**	90.20 ± 0.65	89.34 ± 0.40	**89.82 ± 0.45**
	Covid Variant Classification	Covid	9	**74.09 ± 0.25**	70.84 ± 0.80	73.40 ± 0.30	64.07 ± 0.51	**72.45 ± 0.21**
	Species Classification	Virus	20	67.24 ± 2.10	59.17 ± 3.32	**77.12 ± 1.23**	71.34 ± 2.05	**77.21 ± 1.04**
		Fungi	25	15.06 ± 1.32	18.75 ± 1.77	**30.01 ± 0.76**	15.07 ± 1.88	**30.14 ± 2.48**

Table 6. Fit time (s/epoch) on genome sequence learning. The fastest times are in **green** and the slowest ones are in **red**.

Dataset	$\mathbb{P}^n_K$ PMLR	$\mathbb{P}^n_K$ PBMLR-P	$\mathbb{P}^n_K$ BMLR	$\mathbb{L}^n_K$ LMLR	$\mathbb{L}^n_K$ BMLR-L
LTR Copia	3.96	**5.11**	4.06	3.92	**3.77**
LINEs	5.80	**6.90**	5.73	5.50	**5.32**
SINEs	1.36	**1.80**	1.39	1.38	**1.28**
CMC-EnSpm	3.11	**4.38**	3.04	2.89	**2.82**
hAT-Ac	4.37	**5.37**	4.38	4.13	**3.96**
processed	5.02	**5.94**	4.89	4.68	**4.58**
unprocessed	3.30	**3.90**	3.29	3.05	**2.95**
CPD-tata	0.72	**1.35**	0.70	0.67	**0.62**
CPD-notata	6.13	**12.40**	6.00	5.73	**5.71**
CPD-all	6.85	**14.35**	6.59	6.35	**6.33**
PD-tata	0.98	**1.20**	0.97	1.04	**0.94**
PD-notata	8.33	**10.53**	8.29	8.11	**7.84**
PD-all	9.40	**12.07**	9.19	9.06	**8.83**
Covid	28.96	**45.52**	27.97	27.58	**26.67**
Virus	25.28	**29.57**	25.67	25.12	**24.85**
Fungi	6.41	**8.96**	6.41	**6.25**	**6.25**

5.2. Genome sequence learning

Setup. Recently, Khan et al. [33] demonstrated the effectiveness of hyperbolic embedding in genome sequence learning. Following their settings, we adopt a CNN backbone, which consists of three convolutional blocks and an MLR head. Similar to Sec. 5.1, we compare our BMLR against previous hyperbolic MLR heads by replacing the final Euclidean MLR with a hyperbolic MLR. We validate on two benchmarks: Transposable Element Benchmark (TEB) [33] and Genome Understanding Evaluation (GUE) [62], covering a total of 16 datasets. More details are provided in Sec. E.2.

Results. Tab. 5 summarizes 5-fold average Matthews Correlation Coefficient (MCC) across TEB and GUE. Compared with other hyperbolic MLRs, our BMLR-P and BMLR-L achieve higher MCC in most tasks. Similar to Sec. 5.1, the gains are more pronounced on complex datasets with more classes, *e.g.*, Virus (20 classes) and Fungi (25 classes), demonstrating the effectiveness of our approach. Tab. 6 reports fit time per epoch, where PBMLR-P is consistently the slowest due to batch inefficiency, and BMLR-L is the fastest.

Table 7. Comparison of hyperbolic FC layers on link prediction. Best results within each hyperbolic model are in **bold**.

Space	Method	Methodology	Disease $\delta = 0$	Airport $\delta = 1$	PubMed $\delta = 3.5$	Cora $\delta = 11$
$\mathbb{P}^n_K$	Möbius	Tangent	76.35 ± 1.83	93.31 ± 0.41	**94.93 ± 0.06**	90.80 ± 0.56
	Poincaré FC	Poincaré geometry	79.45 ± 1.01	94.31 ± 0.16	94.24 ± 0.25	88.21 ± 0.72
	BFC-P	Busemann	**80.45 ± 0.93**	**94.88 ± 0.39**	94.85 ± 0.07	**91.94 ± 0.32**
$\mathbb{L}^n_K$	LTFC	Tangent	71.32 ± 5.36	92.68 ± 0.35	94.85 ± 0.17	89.37 ± 0.64
	Lorentz FC	Ambient Minkowski	72.78 ± 2.04	92.99 ± 0.33	94.20 ± 0.10	92.06 ± 0.62
	BFC-L	Busemann	**78.36 ± 0.51**	**95.37 ± 0.17**	**94.90 ± 0.04**	**92.28 ± 0.12**

Table 8. Node classification F1 scores of hyperbolic MLRs on the HGCN backbone, where δ denotes the graph hyperbolicity (lower is more hyperbolic). The best results within each hyperbolic model are highlighted in **bold**.

Space	Method	Disease $\delta = 0$	Airport $\delta = 1$	PubMed $\delta = 3.5$	Cora $\delta = 11$
$\mathbb{P}^n_K$	HGCN	86.87 ± 2.58	85.34 ± 1.16	76.29 ± 0.98	76.56 ± 0.81
	HGCN-PMLR	88.98 ± 1.96	84.78 ± 1.48	76.02 ± 1.09	77.47 ± 1.15
	HGCN-PBMLR-P	89.05 ± 0.78	85.04 ± 0.97	75.89 ± 0.78	77.90 ± 1.00
	HGCN-BMLR-P	**92.45 ± 0.96**	**86.02 ± 0.53**	**77.36 ± 0.73**	**78.48 ± 1.52**
$\mathbb{L}^n_K$	HGCN	87.83 ± 0.77	84.94 ± 1.40	76.49 ± 0.88	77.37 ± 1.72
	HGCN-LMLR	89.72 ± 1.51	82.61 ± 1.01	75.44 ± 1.17	69.91 ± 3.61
	HGCN-BMLR-L	**90.80 ± 1.15**	**85.27 ± 1.17**	**77.30 ± 0.41**	**77.65 ± 2.10**

5.3. Node classification

Setup. Following Nguyen et al. [45], we adopt the HGCN [8] backbone to evaluate our BMLR on graph datasets, including Disease [1], Airport [61], PubMed [44], and Cora [50]. The HGCN backbone consists of a hyperbolic Graph Convolutional Network (GCN) and an MLR as the final classification layer. Both the GCN and the MLR are built on the hyperbolic space. The vanilla HGCN uses a tangent MLR, which maps features into the tangent space via Log_e and applies a Euclidean MLR. We replace this with different hyperbolic MLRs. More details are provided in Sec. E.3.

Results. Tab. 8 reports average F1 scores. Our BMLRs consistently outperform prior hyperbolic MLRs within each hyperbolic model. As graphs become less hyperbolic, that is, for larger δ, existing hyperbolic heads could underperform the vanilla tangent-based MLR, for example, PBMLR-P on PubMed, and LMLR on Airport, PubMed, and Cora. Especially on Cora, which has the largest δ, LMLR lags the tangent baseline by a large margin (69.91 vs. 77.37). In contrast, BMLR remains the top performer across all δ values, indicating that Busemann-based decoding robustly strengthens HGCN over a broader range of graph hyperbolicity.

5.4. Link prediction

Setup. We compare our BFC layers with prior hyperbolic FC layers, including the Möbius layer [23] that operates via the tangent space, the Lorentz FC layer [11] that operates through the ambient Minkowski space, and the Poincaré FC layer [51]. Mimicking the Möbius layer, we further implement a Lorentz tangent FC layer, $\mathrm{Log}_{\overline{\mathbf{0}}}(M\,\mathrm{Log}_{\overline{\mathbf{0}}}(x))$, referred to as LTFC. Following Chami et al. [8], we evaluate on four graph datasets for the link prediction task: Disease [1], Airport [61], PubMed [44], and Cora [50]. Following the HNN implementation [8, 23], all methods share the same backbone with two FC layers. For a fair comparison, all hyperbolic FC layers are followed by a gyroaddition biasing. For BFC, we set $\phi = \tanh$ on Airport and Cora, which yields better performance, while we use the identity map on the other two datasets. More details are provided in Sec. E.4.

Results. Tab. 7 reports 5-fold test AUC. Our BFC layers generally outperform prior hyperbolic FC layers. The gains are most pronounced on Disease, which is the most hyperbolic ($\delta = 0$), where Busemann-based decoding is markedly more effective than tangent or ambient methods, indicating a better capture of intrinsic hyperbolic geometry. This observation aligns with geometric intuition, since tangent space or ambient space approximations inherently struggle to represent curved manifolds in highly non-Euclidean cases. Sec. E.4.2 summarizes fit time and parameter counts. LTFC is the slowest due to costly logarithmic and exponential maps, and LFC uses the largest number of parameters among Lorentz variants. In contrast, our BFC layers achieve training time and model size comparable to those of existing layers.

6. Conclusion

We introduce BMLR and BFC as intrinsic components for hyperbolic neural networks on the Poincaré and Lorentz models, both built from the Busemann function. BMLR provides compact parameters, a point-to-horosphere interpretation, batch-efficient computation, and a limit that recovers Euclidean MLR. BFC extends FC and activation layers with practical $O(nm)$ complexity. Experiments across image classification, genome sequence learning, node classification, and link prediction validate the effectiveness and efficiency of our approaches. BMLR shows increasingly strong performance as the number of classes increases, and replacing existing hyperbolic FC layers with BFC yields additional gains. These results indicate that Busemann geometry offers unified and effective mathematical tools for building hyperbolic neural networks.

Acknowledgements

This work was supported by EU Horizon project ELLIOT (No. 101214398) and by the FIS project GUIDANCE (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 Leonardo at CINECA, Italy.

References

[1] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford University Press, 1991. 8, 21

[2] Gregor Bachmann, Gary Bécigneul, and Octavian Ganea. Constant curvature graph convolutional networks. In *ICML*, 2020. 1

[3] Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. In *ICLR*, 2024. 1, 4, 5, 6, 17, 19

[4] Ahmad Bdeir, Johannes Burchert, Lars Schmidt-Thieme, and Niels Landwehr. Robust hyperbolic learning with curvature-aware optimization. In *NeurIPS*, 2025. 1

[5] Clément Bonet, Lucas Drumetz, and Nicolas Courty. Sliced-Wasserstein distances and flows on Cartan-Hadamard manifolds. *JMLR*, 2025. 1, 2

[6] Martin R Bridson and André Haefliger. *Metric spaces of non-positive curvature*. Springer Science & Business Media, 2013. 1, 2, 13, 15

[7] James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31 (59-115):2, 1997. 2

[8] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, 2019. 1, 8, 21, 22

[9] Ines Chami, Albert Gu, Dat P Nguyen, and Christopher Ré. HoroPCA: Hyperbolic dimensionality reduction via horospherical projections. In *ICML*, 2021. 1

[10] Tianyu Chen, Xingcheng Fu, Yisen Gao, Haodong Qian, Yuecen Wei, Kun Yan, Haoyi Zhou, and Jianxin Li. Galaxy walker: Geometry-aware vlms for galaxy-scale understanding. In *CVPR*, 2025. 1

[11] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *ACL*, 2022. 1, 6, 8, 22

[12] Ziheng Chen, Yue Song, Xiao-Jun Wu, and Nicu Sebe. RMLR: Extending multinomial logistic regression into general geometries. In *NeurIPS*, 2024. 16

[13] Ziheng Chen, Yue Song, Xiao-Jun Wu, and Nicu Sebe. Gyrogroup batch normalization. In *ICLR*, 2025. 1

[14] Ziheng Chen, Xiao-Jun Wu, Bernhard Schölkopf, and Nicu Sebe. Building transformation layers for Riemannian neural networks, 2025. 5

[15] Ziheng Chen, Xiao-Jun Wu, Bernhard Schölkopf, and Nicu Sebe. Riemannian batch normalization: A gyro approach. *arXiv preprint arXiv:2509.07115*, 2025. 1, 3, 15, 16

[16] Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional network. In *CVPR*, 2021. 1

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6, 19

[18] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. Hyperbolic image-text representations. In *ICML*, 2023. 1

[19] Manfredo Perdigao Do Carmo and J Flaherty Francis. *Riemannian Geometry*. Springer, 1992. 12

[20] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *CVPR*, 2022. 1

[21] Xiran Fan, Chun-Hao Yang, and Baba Vemuri. Horospherical decision boundaries for large margin classification in hyperbolic space. In *NeurIPS*, 2023. 1, 4

[22] Xingcheng Fu, Yisen Gao, Yuecen Wei, Qingyun Sun, Hao Peng, Jianxin Li, and Xianxian Li. Hyperbolic geometric latent diffusion model for graph generation. In *ICML*, 2024. 1

[23] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *NeurIPS*, 2018. 1, 4, 5, 6, 8, 15, 17, 22

[24] Zhi Gao, Yuwei Wu, Yunde Jia, and Mehrtash Harandi. Curvature generation in curved spaces for few-shot learning. In *ICCV*, 2021. 1

[25] Zhi Gao, Chen Xu, Feng Li, Yunde Jia, Mehrtash Harandi, and Yuwei Wu. Exploring data geometry for continual learning. In *CVPR*, 2023. 1

[26] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic busemann learning with ideal prototypes. In *NeurIPS*, 2021. 1

[27] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. In *ICLR*, 2019. 1

[28] Yunhui Guo, Xudong Wang, Yubei Chen, and Stella X Yu. Clipped hyperbolic classifiers are super-hyperbolic classifiers. In *CVPR*, 2022. 6, 19

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[30] Neil He, Rishabh Anand, Hiren Madhu, Ali Maatouk, Smita Krishnaswamy, Leandros Tassiulas, Menglin Yang, and Rex Ying. HELM: Hyperbolic large language models via mixture-of-curvature experts. In *NeurIPS*, 2025. 1

[31] Neil He, Menglin Yang, and Rex Ying. Lorentzian residual neural networks. In *KDD*, 2025. 1

[32] Isay Katsman, Eric Chen, Sidhanth Holalkere, Anna Asch, Aaron Lou, Ser Nam Lim, and Christopher M De Sa. Riemannian residual neural networks. In *NeurIPS*, 2024. 1

[33] Raiyan R Khan, Philippe Chlenski, and Itsik Pe'er. Hyperbolic genome embeddings. In *ICLR*, 2025. 1, 7, 20, 21

[34] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *CVPR*, 2020. 1

[35] Diederik P Kingma. Adam: A method for stochastic optimization. In *ICLR*, 2015. 22
[36] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. 6, 19
[37] Yann Le and Xuan Yang. Tiny ImageNet visual recognition challenge, 2015. 6, 19
[38] Guy Lebanon and John Lafferty. Hyperplane margin classifiers on the multinomial manifold. In *ICML*, 2004. 1, 3
[39] John M Lee. *Introduction to Riemannian manifolds*. Springer, 2018. 2
[40] Shanglin Li, Chu Shiwen, Okan Koç, Yi Ding, Qibin Zhao, Motoaki Kawanabe, and Ziheng Chen. HEEGNet: Hyperbolic embeddings for EEG. In *ICLR*, 2026. 1
[41] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *NeurIPS*, 2019. 1
[42] Yuanpei Liu, Zhenqi He, and Kai Han. Hyperbolic category discovery. In *CVPR*, 2025. 1
[43] Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the Fréchet mean. In *ICML*, 2020. 1
[44] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *10th International Workshop on Mining and Learning with Graphs*, page 1, 2012. 8, 21
[45] Xuan Son Nguyen, Shuo Yang, and Aymeric Histace. Neural networks on symmetric spaces of noncompact type. In *ICLR*, 2025. 1, 4, 5, 6, 8, 17, 19, 21, 22
[46] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, 2017. 1
[47] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *ICML*, 2018. 1
[48] Avik Pal, Max van Spengler, Guido Maria D'Amely di Melendugno, Alessandro Flaborea, Fabio Galasso, and Pascal Mettes. Compositional entailment learning for hyperbolic vision-language models. In *ICLR*, 2025. 1
[49] John G Ratcliffe. *Foundations of Hyperbolic Manifolds*. Springer, 2006. 2
[50] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008. 8, 21
[51] Ryohei Shimizu, YUSUKE Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. In *ICLR*, 2021. 1, 4, 5, 6, 8, 17, 22, 24, 25
[52] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *ICLR*, 2020. 3
[53] Yadong Sun, Xiaofeng Cao, Yu Wang, Wei Ye, Jingcai Guo, and Qing Guo. Geometry awakening: Cross-geometry learning exhibits superiority over individual structures. In *NeurIPS*, 2024. 1
[54] Tanuj Sur, Samrat Mukherjee, Kaizer Rahaman, Subhasis Chaudhuri, Muhammad Haris Khan, and Biplab Banerjee. Hyperbolic uncertainty-aware few-shot incremental point cloud segmentation. In *CVPR*, 2025. 1
[55] Abraham Albert Ungar. *Analytic Hyperbolic Geometry and Albert Einstein's Special Theory of Relativity (Second Edition)*. World Scientific, 2022. 3, 15, 16
[56] Max Van Spengler, Erwin Berkhout, and Pascal Mettes. Poincaré resnet. In *ICCV*, 2023. 1
[57] Rui Wang, Yuting Jiang, Xiaoqing Luo, Xiao-Jun Wu, Nicu Sebe, and Ziheng Chen. Wasserstein-aligned hyperbolic multi-view clustering. In *AAAI*, 2026. 1
[58] Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *KDD*, 2024. 1
[59] Menglin Yang, Ram Samarth B B, Aosong Feng, Bo Xiong, Jiahong Liu, Irwin King, and Rex Ying. Hyperbolic fine-tuning for large language models. In *NeurIPS*, 2025. 1
[60] Xin Yang, Xingrun Li, Heng Chang, Xihong Yang, Shengyu Tao, Maiko Shigeno, Ningkang Chang, Junfeng Wang, Dawei Yin, Erxue Min, et al. Hgformer: Hyperbolic graph transformer for collaborative filtering. In *ICML*, 2025. 1
[61] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *NeurIPS*, 2018. 8, 21
[62] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. DNABERT-2: Efficient foundation model and benchmark for multi-species genome. In *ICLR*, 2024. 7, 20