KLEIN NEURAL NETWORKS++

Anonymous authors
Paper under double-blind review

ABSTRACT

Hyperbolic neural networks (HNNs) have shown significant advantages in representing hierarchical and tree-structured data through their exponential volume growth. However, existing methods rely almost exclusively on the Poincaré ball and Lorentz hyperboloid, while the Klein model, another classical representation, has long been unexplored. The Klein model has Euclidean line-segment geodesics and a closed-form Einstein midpoint, where the latter avoids the iterative Fréchet mean required for batch normalization. In this work, we systematically explore its modeling potential and build a Klein neural network framework of closed-form operators, namely Klein MLR, Klein Busemann MLR, Klein fully connected layer, Klein Busemann fully connected layer, and Klein GyroBN. Klein demonstrates strong training stability under feature clipping, curvature mis-specification, and gradient scaling, where the Poincaré and Lorentz references diverge. The closed-form Einstein midpoint in Klein GyroBN offers a favorable trade-off between computational efficiency and performance against iterative Fréchet means. Across numerical stability, image classification, graph link prediction and graph node classification, the proposed Klein layers match or improve upon their Poincaré, Lorentz, and PV counterparts. Klein MLR heads further achieve the fastest runtime among hyperbolic counterparts.

1 INTRODUCTION

Hyperbolic geometry provides a natural representation for hierarchical and tree-like data due to its exponential volume growth, and has driven a growing body of *hyperbolic neural networks* (HNNs) across applications, including computer vision (Khrulkov et al., 2020; Ermolov et al., 2022; Ghadimi Atigh et al., 2022; van Spengler et al., 2023; Bdeir et al., 2024; Pal et al., 2025; Sur et al., 2025; Liu et al., 2025; Bdeir et al., 2025; He et al., 2025b), graph learning (Chami et al., 2019; Bachmann et al., 2020; Fu et al., 2024; Sun et al., 2024; Li et al., 2024), natural language processing and knowledge graphs (Nickel & Kiela, 2017; Ganea et al., 2018; Nickel & Kiela, 2018; Gulcehre et al., 2019; Balazevic et al., 2019; Yang et al., 2025a; 2024; He et al., 2025a), multimodal vision-language learning (Desai et al., 2023), recommendation systems (Yang et al., 2025b), astronomy (Chen et al., 2025a), and genomic sequence learning (Zhou & Sharpee, 2021; Khan et al., 2025). These successes have driven work on hyperbolic space, principally on the Poincaré ball and the Lorentz hyperboloid with a recent extension to the Proper Velocity (PV) model (Chen et al., 2026b). The Poincaré ball and Lorentz hyperboloid have dominated recent work since they offer convenient Riemannian or gyrovector structures (Ganea et al., 2018; Chen et al., 2025b; Ungar, 2022) that facilitate neural network construction. However, the Poincaré ball and Lorentz hyperboloid are constrained spaces and can suffer from numerical instabilities. In particular, as Poincaré-ball embeddings approach the boundary, numerical computations become unstable and may cause vanishing gradients (Guo et al., 2022).

The Klein ball is the third canonical hyperbolic model alongside the Poincaré ball and the Lorentz hyperboloid, originating from Einstein's special relativity (Ungar, 2022). It admits an Einstein gyrovector space, paralleling the Möbius and proper-velocity gyrovector spaces of the Poincaré and PV models (Chen et al., 2026b). Among the three models, only the Klein ball admits Euclidean line segments as geodesics together with a closed-form Einstein midpoint (Ungar, 2022, Ch. 6). These properties make the Klein ball a natural geometry for batch-efficient hyperbolic neural networks, providing a closed-form alternative to the iterative Fréchet means used for aggregation on the Poincaré ball and the Lorentz hyperboloid (Lou et al., 2020; Bdeir et al., 2024). This weighted average recurs throughout hyperbolic architectures as a batch center, few-shot prototype, message aggregator, and

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

attention value mixture. Prior work already uses its Klein form in hyperbolic attention, while the Lorentzian centroid used in Lorentz batch normalization is the same average under an isometry (Gulcehre et al., 2019; Law et al., 2019; Bdeir et al., 2024). Our native derivation removes coordinate conversion within KNN++, while the same closed-form average remains usable in Poincaré and Lorentz pipelines without adopting the Klein model. Beyond these auxiliary uses (Gulcehre et al., 2019; Shimizu et al., 2021; Chen et al., 2025b), recent work has begun to study the Klein model directly. Mao et al. (2024) develops a single MLR layer for graph node classification through Euclidean origin-tangent Exp/Log maps, whereas Zhang et al. (2026) introduces an Einstein-gyrovector metric-learning method for fine-grained image classification. Neither study develops a unified, intrinsic layer framework with Klein-native geodesic and Busemann classifiers, fully connected counterparts, and batch normalization.

In this work, we propose Klein Neural Networks++ (KNN++), a systematic intrinsic Klein-based HNN framework. We instantiate five intrinsic closed-form layers, Klein MLR, BMLR, FC, BFC, and GyroBN, via a Klein hyperplane distance, the Klein Busemann function, and an Einstein-midpoint acceleration of GyroBN. The contribution of KNN++ lies at the framework level. Most individual layers have counterparts in isometric hyperbolic models, but their coordinate expressions differ in computational cost and numerical behavior. In the Klein model, straight-line geodesics and Einstein gyrovector operations yield closed-form, batch-efficient implementations of the complete layer set, including a single-pass batch center (Sec. C.4). KNN++ unifies these operators into an intrinsic toolkit whose computational efficiency and numerical operating range are evaluated below. Klein exhibits stable training behavior under feature clipping, curvature mis-specification, and gradient scaling, in ranges where its Poincaré and Lorentz counterparts diverge. Ablations on the GyroBN gyromean show that the closed-form Einstein midpoint offers a favorable trade-off between speed and accuracy against iterative Fréchet means. Across numerical stability, image classification, link prediction and node classification, KNN++ matches or improves on its Poincaré, Lorentz, and PV counterparts, and the MLR head ablation shows Klein heads are competitive in runtime. Our main contributions are summarized as follows.

1. We develop a new systematic toolkit of fundamental building blocks on the Klein ball, including MLR, Busemann MLR (BMLR), FC, Busemann FC (BFC), and Klein GyroBN.
2. We show that the Klein layers train at every curvature from $|K|$=0.01 to 10 and every clipping radius from 1 to 10^9, while the Poincaré reference diverges in six of these ten configurations and the Lorentz reference in four.
3. We derive a closed-form Klein batch normalization that bypasses the iterative Fréchet means of prior gyrogroup BN via a single-pass Einstein midpoint, while matching the accuracy of converged Fréchet means.
4. We validate the framework on numerical stability, image classification, link prediction, node classification, and ablations on the MLR head and the GyroBN gyromean, where KNN++ matches or improves on its Poincaré, Lorentz, and PV counterparts.

2 PRELIMINARIES

Riemannian geometry (Lee, 2018). Throughout, $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product and $\| \cdot \|$ the induced norm. A *Riemannian manifold* $(\mathcal{M}, g)$ is a smooth manifold equipped with a smoothly varying inner product g_x on each tangent space $T_x\mathcal{M}$. We use Exp_x and Log_x for the exponential map at x and its local inverse, $\mathrm{PT}_{x \to y}$ for the parallel transport along the geodesic from x to y, and $d(x, y)$ for the geodesic distance.

Busemann function (Bridson & Haefliger, 1999). Let γ be a geodesic ray and let $d(\cdot, \cdot)$ denote the geodesic distance. The *Busemann function* associated with γ is defined as

$$\forall x \in \mathcal{M}, \quad B^\gamma(x) = \lim_{t \to \infty} \big(d(x, \gamma(t)) - t \big), \tag{1}$$

and the limit exists on every hyperbolic manifold. 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 with the inner product up to sign. Hence the Busemann function is the intrinsic generalization of the inner product to manifolds.

Horosphere (Bridson & Haefliger, 1999). 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 and a hyperbolic manifold $\mathcal{M}$, the horospheres $H_\tau^v = \{x \in \mathcal{M} \mid B^v(x) = \tau\}$ are equidistant. When $\mathcal{M}$ is Euclidean, $B^v(x) = -\langle x, v\rangle$ recovers the linear hyperplane (with τ negated), so horospheres are the curved counterpart of linear half-space boundaries.

Klein space (Ungar, 2022). Hyperbolic space of constant curvature $K < 0$ admits several equivalent models, the most popular of which are the Poincaré ball $\mathbb{P}_K^n$ and the Lorentz hyperboloid $\mathbb{L}_K^n$. The *Klein model* $\mathbb{K}_{\mathrm{K}}^n = \{x \in \mathbb{R}^n \mid \|x\|^2 < -1/K\}$ (Ungar, 2022, Ch. 3.8) originated in special relativity, parameterizing the relativistic velocity ball with $1/\sqrt{-K}$ as the speed of light. Its Riemannian metric is

$$g_x^{\mathbb{K}}(u, v) = \frac{\langle u, v\rangle}{1 + K\|x\|^2} - \frac{K\,\langle x, u\rangle\langle x, v\rangle}{(1 + K\|x\|^2)^2}, \qquad u, v \in T_x\mathbb{K}_{\mathrm{K}}^n, \tag{3}$$

and we write $\gamma_x = 1/\sqrt{1 + K\|x\|^2}$ for the *Lorentz factor*. Unlike the conformal Poincaré metric, $g_x^{\mathbb{K}}$ has a non-isotropic component along x, but Klein geodesics are Euclidean line segments inside the open ball (Ungar, 2022), in contrast to the circular arcs of $\mathbb{P}_K^n$ and the hyperbolic sections of $\mathbb{L}_K^n$, see Fig. 1. The Riemannian isometries connecting the three models appear in Sec. C.4.1.

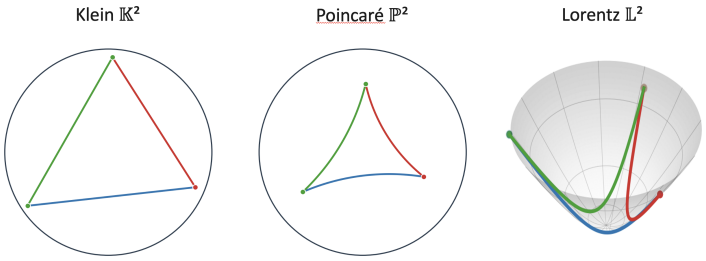


Figure 1: Geodesic triangle on the three hyperbolic models (K=−1).

Klein gyrovector (Ungar, 2022). From an algebraic point of view, the Klein space forms a gyrovector space, which extends the Euclidean vector space to manifolds. Given $x, y \in \mathbb{K}_{\mathrm{K}}^n$ and $t \in \mathbb{R}$, Einstein gyroaddition $\oplus_{\mathrm{E}}$ and scalar gyromultiplication $\otimes_{\mathrm{E}}$ are defined as

$$x \oplus_{\mathrm{E}} y = \frac{1}{1 - K\langle x, y\rangle}\left(x + \tfrac{1}{\gamma_x}\, y - \tfrac{K\gamma_x}{1+\gamma_x}\langle x, y\rangle\, x\right), \tag{4}$$

$$t \otimes_{\mathrm{E}} x = \tfrac{1}{\sqrt{-K}}\,\tanh\!\big(t \operatorname{arctanh}(\sqrt{-K}\|x\|)\big)\,\tfrac{x}{\|x\|}, \quad (x \neq \mathbf{0}), \;\; t \otimes_{\mathrm{E}} \mathbf{0} = \mathbf{0}. \tag{5}$$

In particular, the Einstein inverse is $\ominus_{\mathrm{E}} x = -x$, and the Einstein identity is the zero vector: $\mathbf{0} \oplus_{\mathrm{E}} x = x \oplus_{\mathrm{E}} \mathbf{0} = x$.

A summary of the manifold, metric, distance, exponential and logarithmic maps, and parallel transport on all three hyperbolic models, and detailed reviews of Riemannian geometry, the Busemann function and horospheres, gyrovector spaces, and the Poincaré ball and Lorentz hyperboloid models, are deferred to Sec. C, with a side-by-side summary in Tab. 11.

3 Klein Neural Networks

With the Einstein gyrovector and the Klein Riemannian metric, this section instantiates the core layers of a Klein neural network, namely Multinomial Logistic Regression (MLR), Busemann MLR (BMLR), Fully Connected (FC) and Busemann FC (BFC) maps, and batch normalization. We refer to this layer set as *Klein Neural Networks++* (KNN++), following the progression from HNN to HNN++ (Shimizu et al., 2021). While Mao et al. (2024) introduce a tangent-space Klein classifier, KNN++ provides a set of layers defined intrinsically on $\mathbb{K}_{\mathrm{K}}^n$ for constructing full neural architectures.

3.1 Klein Multinomial Logistic Regression

As shown by Lebanon & Lafferty (2004); Ganea et al. (2018); Chen et al. (2024), the Euclidean MLR $\mathrm{Softmax}(Ax + b)$ can be recast as a signed-margin layer

$$p(y = k \mid x) \propto \exp(v_k(x)), \quad v_k(x) = \operatorname{sign}(\langle a_k, x - p_k\rangle)\,\|a_k\|\,d(x, H_{a_k, p_k}), \tag{6}$$

with $H_{a_k,p_k} = \{x \in \mathbb{R}^n \mid \langle a_k, x - p_k\rangle = 0\}$. Following the Poincaré MLR (Ganea et al., 2018), we define the *Klein hyperplane* parameterized by a base point $p \in \mathbb{K}^n_{\mathrm{K}}$ and a normal direction $a \in T_p\mathbb{K}^n_{\mathrm{K}}$ as the level set $\{x \in \mathbb{K}^n_{\mathrm{K}} \mid \langle \mathrm{Log}^{\mathbb{K}}_p(x), a\rangle_p = 0\}$. The following theorem gives an equivalent gyrovector form together with the point-to-hyperplane distance.

Theorem 3.1 (Klein hyperplane and distance). *[↓] For $x, p \in \mathbb{K}^n_{\mathrm{K}}$ and $a \in T_p\mathbb{K}^n_{\mathrm{K}}$, the Klein hyperplane admits the equivalence*

$$H_{a,p} = \{x \in \mathbb{K}^n_{\mathrm{K}} \mid \langle \mathrm{Log}^{\mathbb{K}}_p(x), a\rangle_p = 0\} = \{x \in \mathbb{K}^n_{\mathrm{K}} \mid \langle (-p) \oplus_{\mathrm{E}} x, w\rangle = 0\}, \tag{7}$$

and the point-to-hyperplane distance is

$$d^{\mathbb{K}}(y, H_{a,p}) = \inf_{q \in H_{a,p}} d^{\mathbb{K}}(y, q) = \frac{1}{\sqrt{-K}} \operatorname{arcsinh}\left(\frac{\sqrt{-K}\,\gamma_z\,|\langle z, w\rangle|}{\|w\|}\right), \tag{8}$$

where $z = (-p) \oplus_{\mathrm{E}} y$, $w = \mathrm{PT}^{\mathbb{K}}_{p\to\mathbf{0}}(a)$, and $\gamma_z = 1/\sqrt{1 + K\|z\|^2}$.

By Thm. 3.1, we define the C-class Klein MLR as

$$p(y{=}k \mid x) \propto \exp(v_k(x)), \quad v_k(x) = \operatorname{sign}\big(\langle (-p_k) \oplus_{\mathrm{E}} x, w_k\rangle\big)\, \|a_k\|_{p_k}\, d^{\mathbb{K}}\big(x, H_{a_k,p_k}\big), \tag{9}$$

parameterized by $p_k \in \mathbb{K}^n_{\mathrm{K}}$ and $a_k \in T_{p_k}\mathbb{K}^n_{\mathrm{K}}$ for each class k. This raw form has three drawbacks. It is over-parameterized, since the bias is absorbed into p_k. It couples the gyroaddition $(-p_k) \oplus_{\mathrm{E}} x$ across classes. It also constrains p_k to $\mathbb{K}^n_{\mathrm{K}}$, requiring Riemannian optimization. To eliminate these drawbacks we follow Shimizu et al. (2021) and adopt the unidirectional parameterization $p_k = \mathrm{Exp}^{\mathbb{K}}_{\mathbf{0}}(r_k\hat{z}_k)$, $a_k = \mathrm{PT}^{\mathbb{K}}_{\mathbf{0}\to p_k}(z_k)$ with $z_k \in \mathbb{R}^n$, $r_k \in \mathbb{R}$, $\hat{z}_k = z_k/\|z_k\|$, which collapses the margin of Eq. (9) into a closed form.

Theorem 3.2 (Klein MLR). *[↓] For $x \in \mathbb{K}^n_{\mathrm{K}}$, the Klein MLR score for class k is*

$$v_k(x) = \frac{\|z_k\|}{\sqrt{-K}} \operatorname{arcsinh}\left(\gamma_x\left(\cosh(\sqrt{-K}\,r_k)\frac{\sqrt{-K}}{\|z_k\|}\langle x, z_k\rangle - \sinh(\sqrt{-K}\,r_k)\right)\right), \tag{10}$$

with $\gamma_x = 1/\sqrt{1 + K\|x\|^2}$. As $K \to 0^-$, $v_k(x) \to \langle x, z_k\rangle - r_k\|z_k\|$, recovering the Euclidean MLR of Eq. (6).

Thm. 3.2 provides a closed-form geodesic Klein MLR via the signed point-to-Klein-hyperplane distance of Eq. (8). Unlike the tangent-space head of Mao et al. (2024), it classifies with the Klein distance itself. The unidirectional parameterization (z_k, r_k) is essential for batchability: Eq. (10) depends on x only through the inner product $\langle x, z_k\rangle$ and the scalar $\|x\|^2$ (via γ_x), implementable as a single matmul XZ, in contrast to the per-class $-p_k \oplus_{\mathrm{E}} x$ tensor that the raw Eq. (9) would require. This reparameterization preserves the full family of Klein geodesic hyperplanes. Indeed, setting $v_k(x) = 0$ in Eq. (10) gives $\langle x, \hat{z}_k\rangle = \tanh(\sqrt{-K}r_k)/\sqrt{-K}$, where $\hat{z}_k$ determines the hyperplane orientation and r_k its signed offset. Hence every Klein geodesic hyperplane, including oblique ones, is represented by some $(\hat{z}_k, r_k)$. Sec. E.1 gives the exact conversion from the raw (p_k, a_k) form and compares the two parameterizations empirically. The full derivation is in Sec. F.2.

3.2 Klein Busemann MLR

The Euclidean signed margin $v_k(x) = \operatorname{sign}(\langle a_k, x - p_k\rangle)\,\|a_k\|\,d(x, H_{a_k,p_k})$ of Eq. (6) is a point-to-hyperplane distance. Chen et al. (2026a) equivalently write this as a point-to-horosphere distance through the Busemann function, see Eq. (1), since horospheres recover Euclidean hyperplanes in the flat limit. Following the horosphere-based MLR of Chen et al. (2026a), we replace geodesic hyperplanes by horospheres as decision boundaries and define the C-class Klein Busemann MLR

$$p(y{=}k \mid x) \propto \exp(u_k(x)), \quad u_k(x) = -\alpha_k B^{v_k}_{\mathbb{K}}(x) + b_k, \tag{11}$$

with learnable $\alpha_k > 0$, $v_k \in \mathbb{S}^{n-1}$, $b_k \in \mathbb{R}$. Evaluating Eq. (11) requires the Klein Busemann function $B^v_{\mathbb{K}}$ in closed form, which the following theorem provides.

Theorem 3.3 (Klein Busemann function). *[↓] Let $v \in \mathbb{S}^{n-1}$ be a unit direction at the origin and let $\gamma(t) = \mathrm{Exp}^{\mathbb{K}}_{\mathbf{0}}(tv)$ be the unit-speed Klein geodesic ray it generates. The Busemann function $B^v_{\mathbb{K}} : \mathbb{K}^n_{\mathrm{K}} \to \mathbb{R}$ associated with γ admits the closed form*

$$B^v_{\mathbb{K}}(x) = \frac{1}{\sqrt{-K}} \log\left(\frac{1 - \sqrt{-K}\langle x, v\rangle}{\sqrt{1 + K\|x\|^2}}\right), \qquad x \in \mathbb{K}^n_{\mathrm{K}}. \tag{12}$$

Table 1: Comparison of hyperbolic MLR methods. In Dist, Real denotes the geodesic point-to-hyperplane distance, while Pseudo denotes a Euclidean surrogate. Compact params indicate whether additional manifold-valued parameters are avoided. Lower FLOPs are better. See Sec. E.1 for details.

Method	**Logit** $u_k(x)$, $\forall k \in \{1,\dots,C\}$	**Space**	**Dist**	**#Params**	**Compact params**	**FLOPs**	**Batch eff.**
Euclidean MLR	$\langle a_k, x\rangle + b_k$ $a_k \in \mathbb{R}^n,\ b_k \in \mathbb{R}$	$\mathbb{R}^n$	Real	$C(n+1)$	✓	$C(2n)$	✓
Poincaré MLR (Ganea et al., 2018)	$\frac{\lambda_{p_k}^K \|a_k\|}{\sqrt{-K}} \operatorname{arcsinh}(\alpha)$ $\alpha = \frac{2\sqrt{-K}\,\langle -p_k \oplus_M x, a_k\rangle}{(1+K\|-p_k\oplus_M x\|^2)\|a_k\|}$ $p_k \in \mathbb{P}_K^n,\ a_k \in T_{p_k}\mathbb{P}_K^n$	$\mathbb{P}_K^n$	Real	$C(2n)$	✗	$C(19n+27)$	✗
Poincaré MLR (Shimizu et al., 2021)	$\frac{2\alpha_k}{\sqrt{-K}} \operatorname{arcsinh}\big(\alpha-\beta\big)$ $\alpha = \lambda_x^K \sqrt{-K}\langle x, v_k\rangle \cosh(2\sqrt{-K}b_k)$ $\beta = (\lambda_x^K - 1)\sinh(2\sqrt{-K}b_k)$ $\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1},\ b_k \in \mathbb{R}$	$\mathbb{P}_K^n$	Real	$C(n+2)$	✓	$C(2n+46)$	✓
Pseudo-Busemann MLR (Nguyen et al., 2025)	$-d(x, p_k)\,\frac{B^{v_k}(-p_k\oplus_M x)}{\|-p_k\oplus_M x\|}$ $p_k \in \mathbb{P}_K^n,\ v_k \in \mathbb{S}^{n-1}$	$\mathbb{P}_K^n$	Pseudo	$C(2n)$	✗	$C(18n+30)$	✗
Lorentz MLR (Bdeir et al., 2024)	$\frac{1}{\sqrt{-K}} \operatorname{sign}(\alpha)\,\beta \big\lvert \operatorname{arcsinh}\big(\sqrt{-K}\,\frac{\alpha}{\beta}\big)\big\rvert$ $z_k \in \mathbb{R}^n,\ b_k \in \mathbb{R}$	$\mathbb{L}_K^n$	Real	$C(n+1)$	✓	$C(4n+51)$	✓
BMLR (Chen et al., 2026a)	$-\alpha_k B^{v_k}(x) + b_k$ $\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1},\ b_k \in \mathbb{R}$ *(P, (p_k, v_k) / v_k-unit / L)*	$\mathbb{P}_K^n$ $\mathbb{P}_K^n$ $\mathbb{L}_K^n$	Real	$C(n+2)$	✓	$C(4n+10)$ $C(2n+12)$ $C(2n+12)$	✓
Klein MLR Sec. 3.1	$\frac{\|z_k\|}{\sqrt{-K}} \operatorname{arcsinh}(\gamma_x \alpha)$ $\alpha = \cosh(\sqrt{-K}r_k)\sqrt{-K}\,\|z_k\|^{-1}\langle x, z_k\rangle - \sinh(\sqrt{}$ $z_k \in \mathbb{R}^n,\ r_k \in \mathbb{R}$	$\mathbb{K}_K^n$	Real	$C(n+1)$	✓	$C(2n+14)$	✓
Klein BMLR Sec. 3.2	$-\frac{\alpha_k}{\sqrt{-K}} \log\left(\frac{1-\sqrt{-K}\langle x, v_k\rangle}{\sqrt{1+K\|x\|^2}}\right) + b_k$ $\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1},\ b_k \in \mathbb{R}$	$\mathbb{K}_K^n$	Real	$C(n+2)$	✓	$C(2n+11)$	✓

Its level sets are Klein horospheres (ellipsoids tangent to the boundary sphere $\{\|y\| = 1/\sqrt{-K}\}$ *at* $v/\sqrt{-K}$*), and* $B_{\mathbb{K}}^v(x) \to -\langle v, x\rangle$ *as* $K \to 0^-$.

The closed form Eq. (12) is derived via the Klein–Lorentz isometry and independently cross-checked through the Klein–Poincaré isometry, with both derivations given in Sec. E.1. The dominant operation is $\sqrt{-K}\langle x, v\rangle$, which reduces to a single scaled matmul $\sqrt{-K}XV$ over a batch.

Substituting Thm. 3.3 into Eq. (11) yields the explicit Klein BMLR logit

$$u_k(x) = \frac{\alpha_k}{\sqrt{-K}}\left[\frac{1}{2}\log(1 + K\|x\|^2) - \log(1 - \sqrt{-K}\langle x, v_k\rangle)\right] + b_k. \tag{13}$$

A first-order expansion as $K \to 0^-$ gives $u_k(x) \to \alpha_k\langle v_k, x\rangle + b_k$, recovering the Euclidean MLR exactly. See Sec. F.3. Klein BMLR complements the geodesic MLR of Sec. 3.1 by providing a horosphere-based formulation, and we evaluate both variants. The level sets of a Busemann function form an equidistant family, $d(H_{\tau_1}^v, H_{\tau_2}^v) = |\tau_2 - \tau_1|$. Consequently, $|u_k(x)|/\alpha_k$ equals the geodesic distance from the level set through x to the decision horosphere. This interpretation holds throughout the score range, whereas distinct geodesic hyperplanes do not form an equidistant family. Sec. C.2 summarizes the Euclidean-to-hyperbolic correspondence underlying this construction (Fig. 3).

Discussion. Tab. 1 compares the logit form, parameter count, FLOP cost, and batch efficiency across hyperbolic MLR methods. Klein BMLR attains the lowest per-class FLOP count $C(2n+11)$ among methods with unconstrained parameters, which translates into the fastest fit time in Tab. 26. Per-method derivations and the FLOP-counting convention are in Sec. E.1.1.

The gap follows from the model rather than the implementation. Klein geodesics are Euclidean line segments, so the pair $(z_k, r_k) \in \mathbb{R}^n \times \mathbb{R}$ specifies a decision boundary completely, whereas the same surface on $\mathbb{P}_K^n$ is a sphere meeting the ideal boundary orthogonally and must be anchored at a manifold-valued point, retracted after every update (Fig. 2, Sec. D.4.1).

3.3 KLEIN FULLY CONNECTED AND BUSEMANN FULLY CONNECTED LAYERS

Shimizu et al. (2021) recast the Euclidean FC layer $y = Ax + b$ as m signed-distance equations $\operatorname{sign}(\langle e_k, y\rangle)\, d(y, H_{e_k,\mathbf{0}}) = \langle a_k, x - p_k\rangle$ for $1 \le k \le m$, with e_k the k-th standard basis vector. Chen et al. (2026a) rewrite the inner product as the Busemann score $-\alpha_k B^{v_k}(x) + b_k$, replacing the geodesic signed-distance with a horosphere-based one. We unify both via the Klein analogue

$$\operatorname{sign}(y_k)\, d^{\mathbb{K}}(y, H_{e_k,\mathbf{0}}) = s_k(x), \quad 1 \le k \le m, \tag{14}$$

which defines the Klein FC $\mathcal{F} : \mathbb{K}^n_{\mathrm{K}} \to \mathbb{K}^m_K$, where s_k is the Klein FC logit v_k from Eq. (10) or the Klein BFC logit u_k from Eq. (13), and admits a unified closed form.

Theorem 3.4 (Klein FC and BFC). *[↓] For $x \in \mathbb{K}^n_{\mathrm{K}}$, the output $y = \mathcal{F}(x) \in \mathbb{K}^m_K$ of Eq.* (14) *is*

$$y_k = \frac{\sinh(\sqrt{-K}\, s_k(x))}{\sqrt{-K}\,\sqrt{1 + \sum_{j=1}^m \sinh^2(\sqrt{-K}\, s_j(x))}}, \quad 1 \le k \le m, \tag{15}$$

with

$$s_k(x) = \begin{cases} v_k(x), & \textit{Klein FC}, \\ u_k(x), & \textit{Klein BFC}. \end{cases} \tag{16}$$

Klein FC and Klein BFC use $m(n{+}1)$ and $m(n{+}2)$ parameters, respectively, and both recover the Euclidean FC as $K \to 0^-$. The proof, together with a comparison of FC vs. BFC costs, is in Sec. F.4.

The shared denominator is the Lorentz time component required to project the output back to the Klein ball. Because it rescales all coordinates equally, it preserves every pairwise ratio y_i/y_j. Further details are provided in Sec. E.2.1.

Generalization with activation. The Euclidean FC with activation $y = \sigma(Ax + b)$ applies σ to the linear preactivation $\langle a_k, x\rangle + b_k$, which under the recast above equivalently appears as $\sigma(s_k(x))$ on the RHS of Eq. (14). We extend the Klein FC/BFC by the same substitution, and Eq. (15) becomes

$$y_k = \frac{\sinh\big(\sqrt{-K}\, \sigma(s_k(x))\big)}{\sqrt{-K}\,\sqrt{1 + \sum_{j=1}^m \sinh^2\big(\sqrt{-K}\, \sigma(s_j(x))\big)}}, \quad 1 \le k \le m. \tag{17}$$

Discussion. Each per-output logit reads directly off Tab. 1, and the sinh-normalized Klein output map adds a shared $18m{+}2$ FLOPs. Klein BFC attains the lowest count $2nm{+}29m{+}2$, m FLOPs below Lorentz BFC-L ($2nm{+}30m{+}2$), inheriting from Klein BMLR the same 1-FLOP-per-output Klein-quotient saving aggregated across the m output dimensions. Tab. 38 in Sec. E.2 lays the full comparison out side by side, extending Chen et al. (2026a, Tab. 3) with our Klein FC and Klein BFC.

3.4 KLEIN BATCH NORMALIZATION

Euclidean Batch Normalization (BN) centers a batch using its mean, normalizes its dispersion using the batch variance, and applies learnable scale and bias parameters. GyroBN (Chen et al., 2025b) lifts these operations to manifolds through their gyrogroup counterparts. We instantiate this template on the Klein ball, first with Fréchet statistics and then with a closed-form Einstein midpoint.

For a batch $\{x_i\}_{i=1}^m \subset \mathbb{K}^n_{\mathrm{K}}$, the corresponding Fréchet mean and variance are

$$\mu_{\mathrm{F}} = \mathrm{FM}\big(\{x_i\}_{i=1}^m\big) = \operatorname*{argmin}_{y \in \mathbb{K}^n_{\mathrm{K}}} \frac{1}{m}\sum_{i=1}^m d^{\mathbb{K}}(x_i, y)^2, \qquad v^2 = \frac{1}{m}\sum_{i=1}^m d^{\mathbb{K}}(x_i, \mu_{\mathrm{F}})^2. \tag{18}$$

Substituting Einstein addition and scalar multiplication into the GyroBN template yields the per-sample update

$$\forall i \le m, \quad \tilde{x}_i \leftarrow \overbrace{\beta \oplus_{\mathrm{E}}}^{\text{Biasing}}\left[\overbrace{\frac{\gamma}{\sqrt{v^2+\varepsilon}} \otimes_{\mathrm{E}}}^{\text{Scaling}}\left(\overbrace{(-\mu_{\mathrm{F}}) \oplus_{\mathrm{E}} x_i}^{\text{Centering}}\right)\right], \tag{19}$$

with learnable $\beta \in \mathbb{K}^n_{\mathrm{K}}$ and $\gamma \in \mathbb{R}_+$, recovering Euclidean BN as $K \to 0^-$. As shown by Chen et al. (2025b), Klein GyroBN has the following homogeneity properties:

$$\mu\big(\{\beta \oplus_{\mathrm{E}} x_i\}\big) = \beta \oplus_{\mathrm{E}} \mu\big(\{x_i\}\big), \tag{20}$$

$$\tfrac{1}{m}\sum_{i=1}^m d^{\mathbb{K}}(t \otimes_{\mathrm{E}} x_i, \mathbf{0})^2 = t^2 \cdot \tfrac{1}{m}\sum_{i=1}^m d^{\mathbb{K}}(x_i, \mathbf{0})^2, \tag{21}$$

for all $\beta \in \mathbb{K}^n_{\mathrm{K}}$ and $t \in \mathbb{R}$, so that Eq. (20) guarantees centering and biasing exactly shift the gyromean while Eq. (21) ensures the scaling step exactly controls the dispersion, reproducing the Euclidean BN normalization guarantee under the Klein geodesic metric. The full inference algorithm, which adapts the GyroBN (Chen et al., 2025b) to the Klein ball, is in Alg. 1 of Sec. E.3.

Closed-form acceleration via the Einstein midpoint. The Fréchet mean μ_{F} has no closed form and requires an iterative solver (Lou et al., 2020), slowing training (Bdeir et al., 2024). We substitute μ_{F} in Eq. (19) by the closed-form *Einstein midpoint* (Ungar, 2022)

$$\mu_{\mathrm{E}} = \frac{\sum_{i=1}^{m} \gamma_{x_i} x_i}{\sum_{i=1}^{m} \gamma_{x_i}}, \qquad \gamma_{x_i} = \frac{1}{\sqrt{1 + K\|x_i\|^2}}, \tag{22}$$

which lies in $\mathbb{K}^n_{\mathrm{K}}$ at $O(mn)$ cost and matches the iterative Fréchet mean within seed noise on node classification. The Lorentz weights increase with radius, giving boundary points greater influence on the Einstein midpoint. We use the midpoint for single-pass batch centering and apply feature clipping to control this sensitivity. See Sec. 4.5 for empirical results and Sec. E.3 for the full analysis and engineering considerations.

4 Experiments

We evaluate our Klein neural networks on the experiments below, with all hyperbolic baselines spanning the Poincaré, Lorentz, and PV models. Full setup details are in Sec. D.

4.1 Numerical Stability

Across the two sweeps below, the Poincaré reference diverges in six of ten configurations and the Lorentz reference in four; however, Klein remains stable throughout. We use two ResNet-18 sweeps and a backward-pass diagnostic of $\exp_{\mathbf{0}}$. The sweeps use CIFAR-100 for 100 epochs over three seeds. Each manifold is deployed through its standard reference implementation. Further diagnostics are in Sec. D.1 and Sec. D.7.

Table 2: Curvature sweep (mean±std, 3 seeds).

$\lvert K\rvert$	Klein	Poincaré	Lorentz
0.01	72.51 ± 0.28	NaN	NaN
0.1	72.77 ± 0.11	73.92 ± 0.12	55.44 ± 11.62
0.3	72.99 ± 0.25	73.58 ± 0.16	72.90 ± 0.13
1.0	73.26 ± 0.30	73.13 ± 0.35	72.94 ± 0.10
10.0	70.80 ± 0.09	NaN	72.71 ± 0.30

Curvature magnitude. We sweep $|K| \in \{0.01, 0.1, 0.3, 1.0, 10.0\}$ at fixed r=1.0, spanning three orders of magnitude. Klein remains finite across all five curvatures with low variation across seeds. Poincaré and Lorentz both diverge at the smallest $|K|$, Poincaré also diverges at $|K|$=10, and Lorentz exhibits a high-variance collapse at $|K|$=0.1.

Table 3: Clipping radius sweep (mean±std, 3 seeds).

r	Klein	Poincaré	Lorentz
1.0	73.33 ± 0.03	73.74 ± 0.14	72.91 ± 0.27
2.0	74.61 ± 0.21	NaN	72.13 ± 0.25
5.0	73.98 ± 0.25	NaN	NaN
10.0	73.16 ± 0.73	NaN	NaN
10^9	71.14 ± 0.40	NaN	NaN

Feature clipping radius. We sweep $r \in \{1, 2, 5, 10, 10^9\}$ at fixed K=−0.3. The r=10^9 row effectively removes clipping. Klein remains finite at every radius, whereas Poincaré diverges at r=2 and both baselines diverge for every $r \geq 5$; the failures trace to the Möbius square root that the polynomial Klein γ composition avoids (Sec. C.4.2).

Gradient norm under feature scaling. A controlled probe maps scaled unit tangents through $\exp_0$ and differentiates their squared norm (Sec. D.1). Klein and Poincaré gradients decay smoothly with r, whereas the Lorentz gradient climbs eight orders of magnitude to 1.8×10^6 at r=10 (Tab. 4), the radius range where Tab. 3 also records its training failure, attributing that failure to the coordinate map rather than to the classifier.

Table 4: Mean tangent-input gradient norm.

r	Klein	Poincaré	Lorentz
1	2.5e−3	2.5e−3	2.8e−2
5	1.4e−6	7.6e−11	8.6e+1
10	6.0e−11	3.6e−11	1.78e+6
50	1.2e−11	7.2e−12	3.2e−3
100	6.0e−12	3.6e−12	1.6e−3
500	1.2e−12	7.2e−13	3.1e−4
1000	6.1e−13	3.6e−13	1.6e−4

Together, these experiments mark a capability boundary because baselines can fail away from default scaling or curvature while Klein remains trainable. Bounded coordinates, cancellation-free logits, and closed-form batch centering explain this range. Lorentz matches Klein under composition, and all three collapse at sufficient depth (Secs. D.6.2 to D.6.4 and Sec. D.1).

4.2 Image Classification

Setup. Following Guo et al. (2022), we adopt a ResNet-18 (He et al., 2016) Euclidean backbone and replace only the final classifier, comparing the two Klein heads against a Euclidean baseline and the six hyperbolic heads of Tab. 5 on CIFAR-10/100 (Krizhevsky & Hinton, 2009) over five seeds, and all nine heads on ImageNet-1K (Deng et al., 2009) following the protocol of Chen et al. (2026a). Full training details are in Sec. D.2.

Results. Within the geodesic MLR family, Klein MLR attains the highest accuracy on CIFAR-10, CIFAR-100 and ImageNet-1K (Tab. 5). Both Klein heads also carry Euclidean-scale parameter counts, half that of PBMLR-P, and the lowest runtime among hyperbolic heads at every width (Tab. 26).

Table 5: Hyperbolic MLR accuracy (%) on a ResNet-18 backbone. Best in **bold**, largest #Params in red.

Family	Space	Method	CIFAR-10 (C=10)		CIFAR-100 (C=100)		ImageNet-1K (C=1000)	
			Acc	#Params	Acc	#Params	Top-1	#Params
Euclidean	$\mathbb{R}^n$	Euclidean MLR (He et al., 2016)	95.14±0.12	5.13K	77.72±0.15	51.30K	71.89	513.00K
Pseudo	$\mathbb{P}^n_K$	PBMLR-P (Nguyen et al., 2025)	95.23±0.08	10.24K	77.78±0.15	102.40K	71.57	1024K
Geodesic	$\mathbb{P}^n_K$	PMLR (Shimizu et al., 2021)	95.04±0.13	5.14K	77.19±0.50	51.40K	71.70	514.00K
	$\mathbb{L}^n_K$	LMLR (Bdeir et al., 2024)	94.98±0.12	5.13K	78.03±0.21	51.30K	72.50	513.00K
	$\mathbb{PV}^n_K$	PV MLR (Chen et al., 2026b)	95.22±0.04	5.13K	78.05±0.17	51.30K	72.46	513.00K
	$\mathbb{K}^n_K$	**Klein MLR**	**95.29±0.15**	5.13K	**78.25±0.13**	51.30K	**73.16**	513.00K
Busemann	$\mathbb{P}^n_K$	BMLR-P (Chen et al., 2026a)	95.32±0.14	5.14K	78.10±0.35	51.40K	**73.30**	514.00K
	$\mathbb{L}^n_K$	BMLR-L (Chen et al., 2026a)	95.25±0.02	5.14K	78.07±0.26	51.40K	73.24	514.00K
	$\mathbb{K}^n_K$	**Klein BMLR**	**95.33±0.08**	5.14K	**78.24±0.13**	51.40K	73.20	514.00K

4.3 Link Prediction

Setup. We compare our Klein FC and Klein BFC against prior hyperbolic FC layers, Möbius FC (Ganea et al., 2018) and LTFC (Chen et al., 2026a) (tangent space), Lorentz FC (Chen et al., 2022) (ambient Minkowski space), Poincaré FC (Shimizu et al., 2021), PLFC (Shi et al., 2026) and PV FC (Chen et al., 2026b) (geodesic hyperplanes), and BFC-P/BFC-L (Chen et al., 2026a) (horospheres). Following Chami et al. (2019); Chen et al. (2026b), we evaluate on four graph datasets (Disease (Anderson & May, 1991), Airport (Zhang & Chen, 2018), PubMed (Namata et al., 2012), Cora (Sen et al., 2008)) ordered by Gromov δ-hyperbolicity. All methods share a 2-layer hyperbolic FC encoder and a Fermi–Dirac decoder. More details are in Sec. D.3.

Results. A Klein layer attains the best 5-fold test AUC in every column of Tab. 6, significantly so on Disease and PubMed, and statistically indistinguishable from four other layers on Airport. Klein BFC leads the three more curved splits; on the near-Euclidean Cora the horosphere head drops and geodesic Klein FC stays level with the best baseline.

4.4 Node Classification

Setup. We study node classification on the four graph datasets of Sec. 4.3. All methods share a two-FC-plus-MLR architecture and differ only in the underlying hyperbolic model; the baselines of Shimizu et al. (2021); Bdeir et al. (2024); Shi et al. (2026); Chen et al. (2026b); Mao et al. (2024) are listed in Tab. 21. Our two Klein stacks are KNN++ (Klein FC with Klein MLR) and KBNN++ (Klein BFC with Klein BMLR). More details in Sec. D.3.

Results. Tab. 21 reports 5-fold accuracy. KNN++ attains the highest values on Airport and Cora, and KBNN++ on PubMed, where its improvement over the strongest baseline is significant. Fig. 2 draws that head on Airport in all three models. A decision surface is cheap when it is both totally geodesic and flat in the coordinates the head optimises over, and only Klein gives both at once: the measured bend of the drawn hyperplane is exactly zero there against 0.232 of the ball radius in Poincaré. The bottom row classifies every pixel with that one head, so the panels agree everywhere and differ only in coordinates.

4.5 Ablation on MLR and GyroBN

MLR runtime. Tab. 26 records per-iteration wall time for each MLR head on random Euclidean features at batch 1024, 100 classes and FP32. The measured ordering follows the FLOP ranking of Tab. 1. The Klein variants are the fastest hyperbolic heads at every input dimension, 1.49–1.55 ms against 1.68–2.53 ms for the Poincaré, Lorentz and PV heads. PBMLR-P takes 217.86–222.30 ms, 143–$148\times$ the Klein cost at matched dimension.

GyroBN gyromean. Tab. 7 records accuracy and fit time for five Fréchet truncations, the closed-form Einstein midpoint and a no-BN reference, all on the same Klein FC backbone. The midpoint is

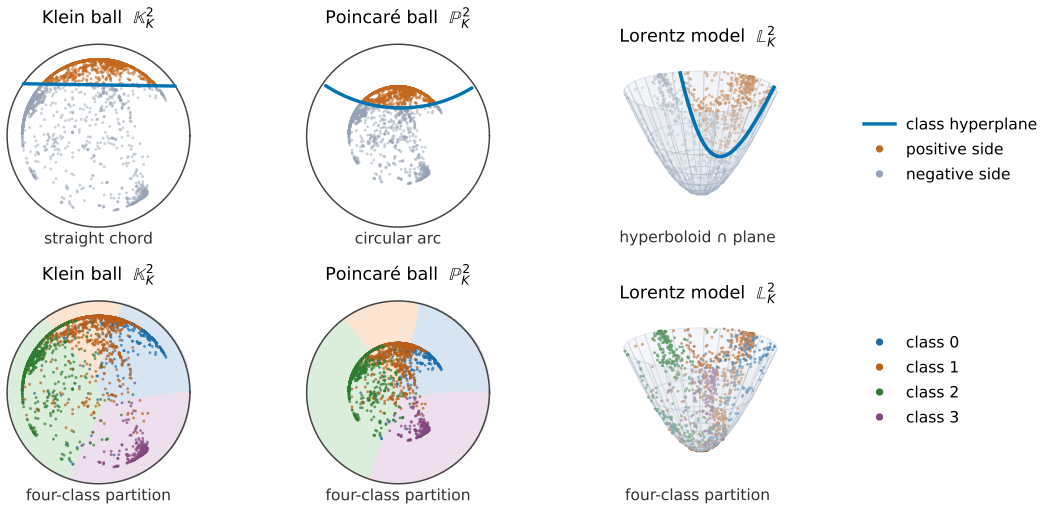


Figure 2: A Klein MLR trained on Airport, seen in the three models ($K=-1$). The stack is that of Sec. 4.4 with the last layer narrowed to two for plotting, reaching 76.7% on the 3188 nodes. Top, one class hyperplane; bottom, the four-class partition.

Table 6: FC/BFC layers on link prediction (ROC AUC %), 5 seeds. Best results are in **bold**.

Space	Method	Disease (δ=0)	Airport (δ=1)	PubMed (δ=3.5)	Cora (δ=11)
$\mathbb{P}^n_K$	Möbius FC	76.35±1.83	93.31±0.41	94.93±0.06	90.80±0.56
	Poincaré FC	79.45±1.01	94.31±0.16	94.24±0.25	88.21±0.72
	BFC-P	80.45±0.93	94.88±0.39	94.85±0.07	90.66±0.39
$\mathbb{L}^n_K$	LTFC	71.32±5.36	92.68±0.35	94.85±0.17	89.37±0.64
	Lorentz FC	75.59±0.77	92.99±0.33	94.61±0.03	92.06±0.62
	PLFC (Shi et al., 2026)	68.92±1.72	95.39±0.13	94.95±0.08	93.47±0.17
	BFC-L	78.36±0.51	94.78±0.59	94.90±0.04	91.07±0.14
$\mathbb{PV}^n_K$	PV FC (Chen et al., 2026b)	73.88±1.06	95.08±0.24	95.29±0.12	94.03±0.14
$\mathbb{K}^n_K$	Klein FC	73.10±4.02	94.54±0.12	94.69±0.23	**94.09±0.28**
$\mathbb{K}^n_K$	**Klein BFC**	**82.18±0.39**	**95.47±0.36**	**95.99±0.13**	92.77±0.43

Table 7: Gyromean methods inside the Klein GyroBN layer. Fastest fit time (ms/epoch) in green, slowest in red.

Method	Disease		Airport		PubMed		Cora	
	Acc	Fit Time	Acc	Fit Time	Acc	Fit Time	Acc	Fit Time
KNN++ (no BN)	81.15±0.26	44.0	97.46±0.91	112.7	73.32±0.78	7.6	51.80±2.18	14.1
Fréchet 1 iter	80.96±0.40	101.2	**98.09±0.57**	274.0	72.44±1.22	16.4	57.80±0.86	31.7
Fréchet 2 iters	80.77±0.21	121.6	96.83±0.65	335.2	72.72±1.39	19.5	57.86±0.77	37.5
Fréchet 5 iters	**81.24±0.71**	184.9	92.73±3.62	546.9	72.14±0.87	28.0	**58.06±1.46**	54.9
Fréchet 10 iters	80.77±0.40	290.9	89.40±3.12	782.0	72.90±0.87	41.9	57.16±0.70	88.7
Fréchet ∞	80.77±0.71	396.2	85.67±7.99	3116.4	**73.08±0.68**	25.9	57.60±0.37	59.4
Einstein midpoint	80.57±0.26	**81.1**	97.59±0.68	**221.3**	72.50±0.64	**13.9**	58.04±0.43	**26.2**

the fastest on all four datasets, $1.9\times$ to $14\times$ ahead of the converged Fréchet mean, and matches its accuracy on three. On Airport the Fréchet mean degrades as it converges, from 98.09 ± 0.57 at one iteration to 85.67 ± 7.99 at convergence, while the midpoint reaches 97.59 ± 0.68 in a single pass with no iteration count to choose. Against the unnormalized backbone it adds 6.2 points on Cora and under a point elsewhere.

5 Conclusion

This work introduces Klein Neural Networks++ (KNN++). From three Klein-native primitives, the signed point-to-hyperplane distance, the Klein Busemann function and the Einstein midpoint, we derive five closed-form layers, MLR, BMLR, FC, BFC and GyroBN. In the two stability sweeps the Klein layers finish all ten configurations, while the Poincaré reference diverges in six of them and the Lorentz reference in four. The closed-form Einstein midpoint replaces iterative Fréchet means at a fraction of their cost. Across three tasks they match their Poincaré, Lorentz and PV counterparts as isometry predicts, at the lowest runtime of any hyperbolic head. Future work extends them to residual networks and Transformers (van Spengler et al., 2023; He et al., 2025b; Yang et al., 2024).

AI USE STATEMENT

We used generative AI tools only for language polishing and for translating mathematical expressions we had already derived into code, not for research ideas, derivations, or experiments. All AI-assisted output was reviewed by the authors, and we take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies layer-level operators for hyperbolic neural networks and raises no concerns under the ICLR Code of Ethics. All experiments use publicly available standard benchmarks, and the work involves no human subjects and no newly released data.

REPRODUCIBILITY STATEMENT

The five proposed layers are given in closed form in Sec. 3 and proved in Sec. F, so the architecture can be reimplemented directly from the paper text. Sec. D reports the full experimental protocol, including the training recipes, the per-dataset hyperparameters, the seed counts and dataset splits used throughout, and the hardware, together with the significance tests behind the reported comparisons and the per-seed numbers they are computed from.

REFERENCES

Roy M. Anderson and Robert M. May. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, 1991.

Gregor Bachmann, Gary Bécigneul, and Octavian-Eugen Ganea. Constant curvature graph convolutional networks. In *ICML*, 2020.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. Multi-relational Poincaré graph embeddings. In *NeurIPS*, 2019.

Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. In *ICLR*, 2024.

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

Martin R. Bridson and André Haefliger. *Metric Spaces of Non-Positive Curvature*. Springer, 1999.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, 2019.

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*, 2025a.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *ACL*, 2022.

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

Ziheng Chen, Xiao-Jun Wu, Bernhard Schölkopf, and Nicu Sebe. Riemannian batch normalization: A gyro approach. *arXiv:2509.07115*, 2025b.

Ziheng Chen, Bernhard Schölkopf, and Nicu Sebe. Hyperbolic busemann neural networks. *arXiv:2602.18858*, 2026a.

Ziheng Chen, Zihan Su, Bernhard Schölkopf, and Nicu Sebe. Proper velocity neural networks. In *ICLR*, 2026b.

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.

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

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

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.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *NeurIPS*, 2018.

Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *CVPR*, 2022.

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

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

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

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*, 2025a.

Neil He, Menglin Yang, and Rex Ying. Lorentzian residual neural networks. In *KDD*, 2025b.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Raiyan R. Khan, Philippe Chlenski, and Itsik Pe'er. Hyperbolic genome embeddings. In *ICLR*, 2025.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Marc T. Law, Renjie Liao, Jake Snell, and Richard S. Zemel. Lorentzian distance learning for hyperbolic representations. In *ICML*, 2019.

Guy Lebanon and John Lafferty. Hyperplane margin classifiers on the multinomial manifold. In *ICML*, 2004.

John M. Lee. *Introduction to Riemannian Manifolds*. Springer, 2nd edition, 2018.

Yancong Li, Xiaoming Zhang, Ying Cui, and Shuai Ma. Hyperbolic graph neural network for temporal knowledge graph completion. In *Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, 2024.

Yuanpei Liu, Zhenqi He, and Kai Han. Hyperbolic category discovery. In *CVPR*, 2025.

Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge J. Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the Fréchet mean. In *ICML*, 2020.

Yidan Mao, Jing Gu, Marcus C. Werner, and Dongmian Zou. Klein model for hyperbolic neural networks. In *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, 2024.

Galileo Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *Workshop on Mining and Learning with Graphs (MLG)*, 2012.

Xuan Son Nguyen, Shuo Yang, and Aymeric Histace. Neural networks on symmetric spaces of noncompact type. In *ICLR*, 2025.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NeurIPS*, 2017.

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. In *ICML*, 2018.

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.

Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *ICML*, 2018.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Xianglong Shi, Ziheng Chen, Yunhan Jiang, and Nicu Sebe. Intrinsic lorentz neural network. *arXiv:2602.23981*, 2026.

Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. In *ICLR*, 2021.

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.

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.

Abraham A. Ungar. *Analytic Hyperbolic Geometry and Albert Einstein's Special Theory of Relativity*. World Scientific, second edition, 2022.

Max van Spengler, Erwin Berkhout, and Pascal Mettes. Poincaré ResNet. In *ICCV*, 2023.

B. L. Welch. The generalization of 'student's' problem when several different population variances are involved. *Biometrika*, 34(1–2):28–35, 1947.

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.

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*, 2025a.

Xin Yang, Xingrun Li, Heng Chang, Jinze Yang, Xihong Yang, Shengyu Tao, Maiko Shigeno, Ningkang Chang, Junfeng Wang, Dawei Yin, and Erxue Min. HGformer: Hyperbolic graph transformer for collaborative filtering. In *ICML*, 2025b.

Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *NeurIPS*, 2018.

Yulin Zhang, Boxuan Hu, Huimin Li, and Junlin Hu. Klein hyperbolic metric learning. In *ICML*, 2026.

Yuansheng Zhou and Tatyana O. Sharpee. Hyperbolic geometry of gene expression. *iScience*, 24(3): 102225, 2021.

Klein Neural Networks++

Supplementary Material

APPENDIX CONTENTS

A Notation **15**

B Extended Related Work **16**

B.1 Hyperbolic Representation Learning 16

B.2 Hyperbolic Batch Normalization and Deep Architectures 16

B.3 Hyperbolic Graph Neural Networks 16

B.4 Genome and Biological Sequence Classification 16

C Preliminaries **16**

C.1 Riemannian Geometries 16

C.2 Busemann Function 17

C.3 Gyro-Structures 18

C.4 Three Hyperbolic Models 19

D Experimental Details **22**

D.1 Numerical Stability 23

D.2 Image Classification 25

D.3 Graph Learning 25

D.4 Computational and Geometric Analysis 27

D.5 MLR Head Runtime 28

D.6 Statistical and Robustness Analysis 29

D.7 Learned Representations 30

E Additional Details of Klein Neural Network Layers **31**

E.1 Klein MLR and BMLR 31

E.2 Klein FC and BFC 35

E.3 Klein GyroBN 37

F Proofs **39**

F.1 Proof of Thm. 3.1 39

F.2 Proof of Thm. 3.2 40

F.3 Klein BMLR: derivation of the expanded form 40

F.4 Proof of Thm. 3.4 (Klein FC and BFC) 41

A NOTATION

Tab. 8 summarizes the notation used throughout the paper.

Table 8: Summary of notation.

Notation	Description
Riemannian geometry	
$\mathcal{M}$	Riemannian manifold
$T_x\mathcal{M}$	Tangent space at x
$g_x(u, v)$	Riemannian metric at x
$d(x, y)$	Geodesic distance
$\mathrm{Exp}_x(v)$	Exponential map at x
$\mathrm{Log}_x(y)$	Logarithmic map at x
$\mathrm{PT}_{x\to y}(v)$	Parallel transport of v from x to y
$\gamma(t)$	Unit-speed geodesic ray
$B^\gamma(x)$	Busemann function associated with γ
$B^v(x)$	Busemann function associated with unit direction $v\in\mathbb{S}^{n-1}$
H^v_τ	Horosphere $\{x \mid B^v(x)=\tau\}$
$H_{a,p}$	Hyperbolic hyperplane parameterized by p and tangent normal a
Euclidean	
$\mathbb{R}^n$	Euclidean space of dimension n
$\langle\cdot,\cdot\rangle$, $\|\cdot\|$	Euclidean inner product and norm
$\mathbf{0}$	Zero vector in $\mathbb{R}^n$
$\mathbb{S}^{n-1}$	Unit sphere in $\mathbb{R}^n$
$e_k\in\mathbb{R}^m$	k-th standard basis vector
Hyperbolic models (curvature $K<0$)	
$\mathbb{K}^n_K$	Klein ball model in $\mathbb{R}^n$ with curvature K
$\mathbb{P}^n_K$	Poincaré ball model in $\mathbb{R}^n$ with curvature K
$\mathbb{L}^n_K$	Lorentz hyperboloid model in $\mathbb{R}^{1+n}$ with curvature K
$\mathbb{PV}^n_K$	Proper-velocity manifold with curvature K
$\langle x, y\rangle_{\mathbb{L}}$	Lorentzian inner product $\langle x_s, y_s\rangle - x_t y_t$
$x=[x_t, x_s^\top]^\top$	Time and spatial decomposition in the Lorentz ambient space
Gyrovector operations	
$\oplus_{\mathrm{E}}$, $\otimes_{\mathrm{E}}$	Einstein gyroaddition and scalar gyromultiplication on $\mathbb{K}^n_K$
$\ominus_{\mathrm{E}}$	Einstein inverse
$\oplus_{\mathrm{M}}$, $\otimes_{\mathrm{M}}$	Möbius gyroaddition and scalar gyromultiplication on $\mathbb{P}^n_K$
$\gamma_x=1/\sqrt{1+K\|x\|^2}$	Klein Lorentz factor
$\lambda^K_x=2/(1+K\|x\|^2)$	Poincaré conformal factor
$\mathrm{gyr}[x, y]$	Gyration automorphism induced by x, y
Layers and learning	
C	Number of classes
m, n	Output / input dimension of an FC layer
$u_k(x)$	Logit for class k
$\phi:\mathbb{R}\to\mathbb{R}$	Activation function
$\mathrm{sign}(\cdot)$	Sign function
$\mathrm{softmax}$	Softmax operator
$\mathrm{FM}(\{x_i\})$	Fréchet mean of $\{x_i\}$ on $\mathbb{K}^n_K$
μ_{E}, μ_{F}	Einstein midpoint, Fréchet mean

B Extended Related Work

This appendix extends the related-work discussion in Sec. 1 with additional context on hyperbolic representation learning, batch normalization, graph networks, and genomic applications that we omitted from the main text for space.

B.1 Hyperbolic Representation Learning

The use of hyperbolic geometry for machine learning was pioneered by Nickel & Kiela (2017), who showed that Poincaré embeddings can represent WordNet-scale taxonomies with far fewer dimensions than Euclidean embeddings, owing to the exponential volume growth of hyperbolic space. Nickel & Kiela (2018) extended this approach to the Lorentz model, demonstrating improved optimization stability. Sala et al. (2018) provided a theoretical analysis of the representation tradeoffs, proving that trees with bounded branching factor embed in $\mathbb{P}^2$ with arbitrarily low distortion. Khrulkov et al. (2020) applied hyperbolic embeddings to computer vision, showing that image features exhibit latent hierarchical structure that is better captured in hyperbolic space. These foundational works establish the motivation for our approach. If data is better represented in hyperbolic space, then operating on it with hyperbolic-native neural network layers should be beneficial.

B.2 Hyperbolic Batch Normalization and Deep Architectures

Extending batch normalization to hyperbolic space requires notions of mean, variance, and rescaling on curved manifolds. Lou et al. (2020) proposed differentiating through the Fréchet mean, enabling its use as a centering operation in deep networks. van Spengler et al. (2023) introduced Poincaré ResNet, which combines the Möbius midpoint for batch normalization, Möbius residual connections, and a tangent-space convolution strategy to build deep hyperbolic residual networks. Bdeir et al. (2024) proposed Lorentz batch normalization based on the Lorentz centroid. Recent work on Klein networks (Mao et al., 2024) establishes the underlying gyrostructure but has not explored the Einstein midpoint as a BN centroid. We develop this connection in Sec. 3.4.

B.3 Hyperbolic Graph Neural Networks

Chami et al. (2019) proposed Hyperbolic Graph Convolutional Networks (HGCN), combining graph convolutional layers with hyperbolic geometry via tangent-space message passing and hyperbolic MLR classifiers. HGCN demonstrated significant gains over Euclidean GCNs on link prediction for datasets with strong hierarchical structure (e.g., Disease and Airport). Subsequent work has explored Lorentz graph networks (Chen et al., 2022). Our link prediction experiments follow the HGCN framework, replacing its Poincaré and Lorentz FC and MLR layers with Klein FC, BFC, and BMLR layers to enable a controlled comparison of the underlying geometry.

B.4 Genome and Biological Sequence Classification

Hyperbolic geometry has recently found applications in computational biology, motivated by the tree-like phylogenetic structure of biological sequences. Khan et al. (2025) introduced the Transposable Element Benchmark (TEB), a suite of genomic classification tasks where hyperbolic models should excel due to the hierarchical taxonomy of transposable element families, demonstrating that Poincaré and Lorentz models can improve over Euclidean baselines on TEB tasks, particularly for multi-class problems with deep class hierarchies.

C Preliminaries

This appendix provides the technical background for the main paper. Sec. C.1 reviews Riemannian geometry, Sec. C.3 states the gyrovector-space axioms, and Sec. C.4 collects the Riemannian operators and gyrovector structures of the three canonical hyperbolic models (Klein, Poincaré, Lorentz).

C.1 Riemannian Geometries

For an in-depth discussion on Riemannian geometry, one can refer to Lee (2018).

A *Riemannian manifold* $(\mathcal{M}, g)$ carries a smoothly varying inner product $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ on each tangent space. The induced norm is $\|v\|_x = \sqrt{g_x(v, v)}$. *Geodesics* are constant-speed

curves that locally minimize length. The shortest distance between $x, y \in \mathcal{M}$ is the geodesic distance $\mathrm{d}(x, y)$.

The *exponential map* $\mathrm{Exp}_x : T_x\mathcal{M} \supset \mathcal{V} \to \mathcal{M}$ sends v to $\gamma_{x,v}(1)$, where $\gamma_{x,v}$ is the geodesic with $\gamma(0) = x$, $\dot{\gamma}(0) = v$. Its local inverse is the *logarithmic map* Log_x. *Parallel transport* $\mathrm{PT}_{x\to y} : T_x\mathcal{M} \to T_y\mathcal{M}$ is the isometric transport of tangent vectors along the geodesic from x to y. A smooth map $f : (\mathcal{M}, g) \to (\widetilde{\mathcal{M}}, \tilde{g})$ between Riemannian manifolds is a *Riemannian isometry* if $g_x(u, v) = \tilde{g}_{f(x)}(d_x f(u), d_x f(v))$ for all $u, v \in T_x\mathcal{M}$, where $d_x f$ is the differential at x.

Tab. 9 summarizes the correspondence between Euclidean and Riemannian operations.

Table 9: Euclidean operations and their Riemannian generalizations.

Operation	**Euclidean**	**Riemannian**
Straight line	Straight line	Geodesic
Subtraction	$\overrightarrow{xy} = y - x$	$\overrightarrow{xy} = \mathrm{Log}_x(y)$
Addition	$y = x + \overrightarrow{xy}$	$y = \mathrm{Exp}_x(\overrightarrow{xy})$
Parallel move	$v \to v$	$\mathrm{PT}_{x\to y}(v)$

C.2 Busemann Function

The Busemann function and its level sets (horospheres) are introduced in Sec. 2, Eq. (1). Here we collect the closed-form Busemann functions on the three hyperbolic models. For a unit direction $v \in \mathbb{S}^{n-1}$ at the origin,

$$\mathbb{K}^n_K: \quad B^v_{\mathbb{K}}(x) = \frac{1}{\sqrt{-K}} \log\left(\frac{1 - \sqrt{-K}\langle x, v\rangle}{\sqrt{1 + K\|x\|^2}}\right), \tag{23}$$

$$\mathbb{P}^n_K: \quad B^v_{\mathbb{P}}(x) = \frac{1}{\sqrt{-K}} \log\left(\frac{\|v - \sqrt{-K}\, x\|^2}{1 + K\|x\|^2}\right), \tag{24}$$

$$\mathbb{L}^n_K: \quad B^v_{\mathbb{L}}(x) = \frac{1}{\sqrt{-K}} \log\Big(\sqrt{-K}\,(x_t - \langle x_s, v\rangle)\Big). \tag{25}$$

The Klein expression is derived as Thm. 3.3 in the main paper. The Poincaré and Lorentz expressions follow Chen et al. (2026a) [Prop. 9]. The three forms are mutually consistent under the Klein–Poincaré and Klein–Lorentz isometries of Eqs. (26) to (27), verified explicitly in Sec. E.1.

Horosphere geometry. In each model the level sets of B^v are horospheres $H^v_\tau = \{x \mid B^v(x) = \tau\}$. On $\mathbb{K}^n_K$ and $\mathbb{P}^n_K$ the horospheres are ellipsoids tangent internally to the boundary sphere. On $\mathbb{L}^n_K$ they are intersections of the upper hyperboloid with affine planes parallel to the asymptotic null cone. As $K \to 0^-$, all three Busemann functions converge to $-\langle x, v\rangle$ and the horospheres degenerate to the Euclidean hyperplanes $H^v_\tau = \{x \in \mathbb{R}^n \mid \langle x, v\rangle = \tau\}$.

Horospheres as complementary decision boundaries. Sec. 3.1 and Sec. 3.2 provide geodesic and horospherical decision boundaries, respectively. The Busemann head complements the geodesic formulation through the intrinsic hyperbolic counterpart of the Euclidean inner product.

Euclidean-to-hyperbolic correspondence. Tab. 10 collects the correspondence. The Euclidean MLR logit is $\langle a_k, x\rangle + b_k$; replacing the inner product by its intrinsic counterpart, the Busemann function of Eq. (1), yields Eq. (11) and changes nothing else. This is why Eq. (13) recovers $\alpha_k\langle v_k, x\rangle + b_k$ exactly as $K \to 0^-$ (Sec. F.3): the Busemann head is the deformation of the Euclidean softmax that leaves the flat case intact.

Table 10: Euclidean notions and their intrinsic hyperbolic counterparts, following Chen et al. (2026a) [Tab. 1].

Euclidean	**Hyperbolic**	**In this paper**
straight line	geodesic	Klein: Euclidean line segment
parallel lines	asymptotic geodesic rays	Sec. 2
inner product $\langle x, v\rangle$	Busemann function $-B^v(x)$	Eqs. (1) to (2)
hyperplane $\{\langle x, v\rangle{=}\tau\}$	horosphere $\{B^v(x){=}\tau\}$	Thm. 3.3
parallel hyperplanes, fixed v	*equidistant* horospheres, fixed v	Sec. 2

Metric parallelism. In $\mathbb{R}^n$ the level sets of a linear functional are equidistant, which is what makes the reading "logit = signed margin" meaningful. Passing to $\mathbb{H}^n$, exactly one of the two families retains this. The level sets of a single Busemann function satisfy $d(H^v_{\tau_1}, H^v_{\tau_2}) = |\tau_2 - \tau_1|$ exactly, and the point-to-horosphere distance is the true geodesic distance $|B^v(x) - \tau|$ (Chen et al., 2026a, Thm. 3.3, Cor. 3.4). Hence for Klein BMLR every level set of the logit is again a horosphere of the same family, and $|u_k(x)| = \alpha_k\, d(x, H_{v_k,\alpha_k,b_k})$ with no approximation. By contrast, two distinct geodesic hyperplanes of $\mathbb{H}^n$ are never equidistant: they either intersect or diverge. The level sets $\{x : d(x, H) = c\}$ of a geodesic head's logit are equidistant hypersurfaces, which are not themselves totally geodesic, so only the decision boundary $c = 0$ lies in the intended family. The "curved" appearance of a horosphere in the Klein chart is therefore a property of the chart, not a loss of flatness.

Ideal-boundary geometry. A geodesic hyperplane meets the ideal boundary in an $(n-2)$-sphere, so each of its half-spaces contains an entire boundary cap; a horoball meets the ideal boundary at exactly one ideal point ξ. In hierarchical data a class is a direction of descent, that is an ideal point, so the horosphere family is indexed by the same object as the labels. Fig. 3 illustrates this: both surfaces cut the ray towards ξ at the same point, hence at the same depth, yet the chord's half-space already contains an $80.8°$ arc of the ideal boundary while the horoball touches it only at ξ.

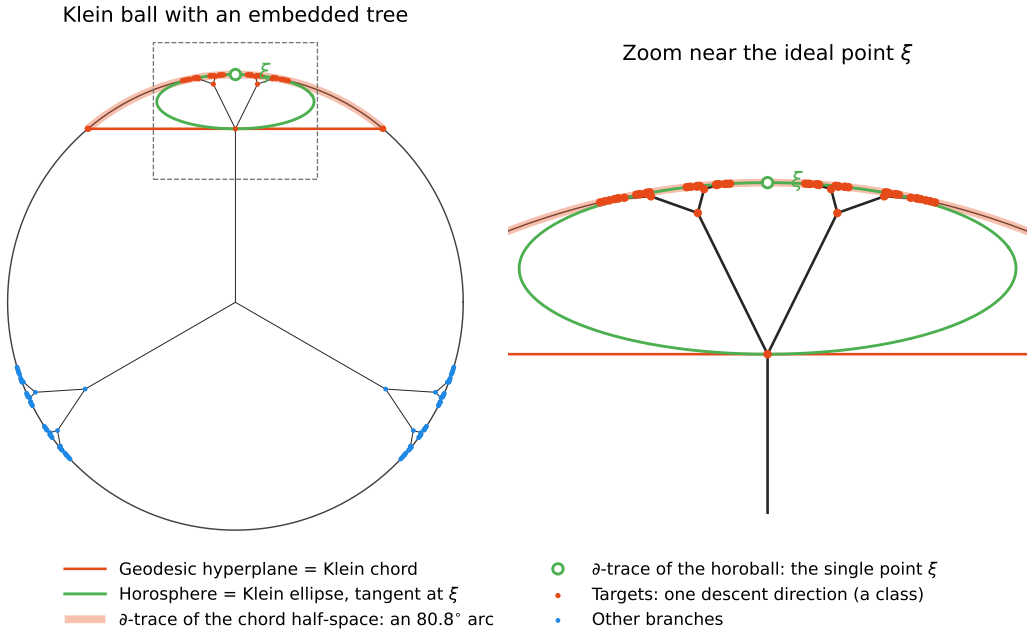


Figure 3: Geodesic and horospherical boundaries at the same depth in the Klein ball.

In Fig. 3, both boundaries intersect the ray towards ξ at the same hyperbolic depth. The geodesic half-space spans a boundary arc, whereas the horoball selects the single ideal direction ξ and encloses its descendants.

C.3 Gyro-Structures

This subsection reviews the gyrovector-space framework of Ungar (2022), which generalizes the vector-space structure to constant-curvature manifolds and underpins the Klein, Poincaré and Lorentz operators of Sec. C.4. Gyrogroups generalize groups. Unlike groups, the binary operation is non-associative but satisfies a left gyroassociative law characterized by a family of automorphisms (gyrations).

Definition C.1 (Gyrogroups (Ungar, 2022))**.** A nonempty set G with binary operation $\oplus : G \times G \to G$ forms a *gyrogroup* $(G, \oplus)$ if its binary operation satisfies the following axioms for any $x, y, z \in G$.

(G1) There is at least one element $e \in G$ called a *left identity* such that $e \oplus x = x$.

(G2) There is an element $\ominus x \in G$ called a *left inverse* of x such that $\ominus x \oplus x = e$.

(G3) There is an automorphism $\mathrm{gyr}[x, y] : G \to G$ for each $x, y \in G$ such that

$$x \oplus (y \oplus z) = (x \oplus y) \oplus \mathrm{gyr}[x, y]z \qquad \text{(Left Gyroassociative Law)}.$$

(G4) Left reduction law $\mathrm{gyr}[x, y] = \mathrm{gyr}[x \oplus y, y]$.

The automorphism $\mathrm{gyr}[x, y]$ is called the *gyration* generated by x, y.

Definition C.2 (Gyrocommutative gyrogroups (Ungar, 2022))**.** A gyrogroup $(G, \oplus)$ is *gyrocommutative* if it satisfies $x \oplus y = \mathrm{gyr}[x, y](y \oplus x)$.

A gyrocommutative gyrogroup equipped with a scalar gyromultiplication satisfying further regularity axioms is a gyrovector space.

Definition C.3 (Gyrovector spaces (Ungar, 2022))**.** A gyrocommutative gyrogroup $(G, \oplus)$ equipped with scalar gyromultiplication $\otimes : \mathbb{R} \times G \to G$ is called a *gyrovector space* if it satisfies the following axioms for any $s, t \in \mathbb{R}$ and $x, y, z \in G$.

(V1) Identity scalar multiplication $1 \otimes x = x$.

(V2) Scalar distributive law $(s + t) \otimes x = s \otimes x \oplus t \otimes x$.

(V3) Scalar associative law $(st) \otimes x = s \otimes (t \otimes x)$.

(V4) Gyroautomorphism property $\mathrm{gyr}[x, y](t \otimes z) = t \otimes \mathrm{gyr}[x, y]z$.

(V5) Identity gyroautomorphism $\mathrm{gyr}[s \otimes x, t \otimes x] = \mathbb{I}$, where $\mathbb{I}$ is the identity map.

Definition C.4 (Real inner-product gyrovector spaces (Ungar, 2022))**.** Let $(G, \oplus, \otimes)$ be a gyrovector space with $G \subseteq \mathbb{R}^n$ inheriting the Euclidean inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$. $(G, \oplus, \otimes, \langle \cdot, \cdot \rangle)$ is a *real inner-product gyrovector space* if it further satisfies the following axioms.

(V6) Inner product gyroinvariance $\langle \mathrm{gyr}[x, y]u, \mathrm{gyr}[x, y]v \rangle = \langle u, v \rangle$ for all $u, v \in G$.

(V7) Scaling preserves direction $\dfrac{|s| \otimes x}{\|s \otimes x\|} = \dfrac{x}{\|x\|}$ for $x \neq \mathbf{0}$ and $s \neq 0$.

(V8) Homogeneity $\|s \otimes x\| = |s| \otimes \|x\|$.

(V9) Gyrotriangle inequality $\|x \oplus y\| \leq \|x\| \oplus \|y\|$.

Definition C.5 (Gyrovector space isomorphism (Ungar, 2022))**.** A bijection $\phi : (G_1, \oplus_1, \otimes_1) \to (G_2, \oplus_2, \otimes_2)$ between two real inner-product gyrovector spaces is a *gyrovector space isomorphism* if for all $x, y \in G_1$ and $t \in \mathbb{R}$, $\phi(x \oplus_1 y) = \phi(x) \oplus_2 \phi(y)$, $\phi(t \otimes_1 x) = t \otimes_2 \phi(x)$, and the inner-product ratio is preserved, $\frac{\langle \phi(x), \phi(y) \rangle}{\|\phi(x)\| \, \|\phi(y)\|} = \frac{\langle x, y \rangle}{\|x\| \, \|y\|}$.

Proposition C.6 (Isomorphism preserves gyrations and identity (Ungar, 2022))**.** *If $\phi : G_1 \to G_2$ is a gyrovector space isomorphism with gyrations gyr_1 and gyr_2, then for all $x, y, z \in G_1$, $\phi(\mathrm{gyr}_1[x, y]z) = \mathrm{gyr}_2[\phi(x), \phi(y)]\phi(z)$, $\phi(e_1) = e_2$, and $\phi(\ominus_1 x) = \ominus_2 \phi(x)$, where e_1, e_2 are the gyro-identities of G_1, G_2.*

The Einstein, Möbius and Lorentz gyrovector spaces are all real inner-product gyrovector spaces, with closed-form operations summarized in Sec. C.4.2.

C.4 Three Hyperbolic Models

C.4.1 Riemannian Operators

The Klein closed-form Riemannian operators in this appendix (distance, exponential and logarithmic maps, parallel transport) follow Chen et al. (2025b), who derive them under arbitrary $K < 0$ via the Klein–Poincaré isometry. We restate them here for completeness and consistent notation with our Klein layer derivations.

There are three standard models of n-dimensional hyperbolic space with sectional curvature $K < 0$.

- **Klein model.** $\mathbb{K}^n_K = \{x \in \mathbb{R}^n \mid \|x\|^2 < -1/K\}$ with metric $g^{\mathbb{K}}_x$ defined in Eq. (3).
- **Poincaré ball.** $\mathbb{P}^n_K = \{x \in \mathbb{R}^n \mid \|x\|^2 < -1/K\}$ with conformal metric $g^{\mathbb{P}}_x = (\lambda^K_x)^2 \langle \cdot, \cdot \rangle$ where $\lambda^K_x = 2/(1 + K\|x\|^2)$.

- **Lorentz model (hyperboloid).** $\mathbb{L}_K^n = \{x = (x_t, x_s) \in \mathbb{R}^{1+n} \mid \langle x, x\rangle_{\mathbb{L}} = 1/K,\ x_t > 0\}$ with Lorentz inner product $\langle x, y\rangle_{\mathbb{L}} = -x_t y_t + \langle x_s, y_s\rangle$.

Tab. 11 summarizes the manifold, metric, distance, exponential and logarithmic maps, and parallel transport on the three models side-by-side.

Table 11: Riemannian operators on the three hyperbolic models ($K < 0$).

Operator	**Klein** $\mathbb{K}_{\mathrm{K}}^n$ (Chen et al., 2025b)	**Poincaré** $\mathbb{P}_K^n$ (Ganea et al., 2018)	**Lorentz** $\mathbb{L}_K^n$ (Nickel & Kiela, 2018)
Manifold	$\{x \in \mathbb{R}^n \mid \|x\|^2 < -1/K\}$	$\{x \in \mathbb{R}^n \mid \|x\|^2 < -1/K\}$	$\{x \in \mathbb{R}^{1+n} \mid \langle x, x\rangle_{\mathbb{L}} = 1/K,\ x_t > 0\}$
$g_x(u, v)$	$\frac{\langle u, v\rangle}{1+K\|x\|^2} - \frac{K\langle x, u\rangle\langle x, v\rangle}{(1+K\|x\|^2)^2}$	$(\lambda_x^K)^2\langle u, v\rangle$	$\langle u_s, v_s\rangle - u_t v_t$
$d(x, y)$	$\frac{1}{\sqrt{-K}}\cosh^{-1}\!\left(\frac{1+K\langle x,y\rangle}{\sqrt{(1+K\|x\|^2)(1+K\|y\|^2)}}\right)$	$\frac{2}{\sqrt{-K}}\tanh^{-1}\!\left(\sqrt{-K}\|-x\oplus_{\mathrm{M}} y\|\right)$	$\frac{1}{\sqrt{-K}}\cosh^{-1}\!\big(K\langle x, y\rangle_{\mathbb{L}}\big)$
$\mathrm{Exp}_x v$	$x + \frac{\sinh(\|v\|_x)\hat{v}}{\cosh(\|v\|_x) + \gamma_x^2\langle x, \hat{v}\rangle\sinh(\|v\|_x)}$	$x \oplus_{\mathrm{M}}\left(\tanh\!\left(\frac{\sqrt{-K}\lambda_x^K\|v\|}{2}\right)\frac{v}{\sqrt{-K}\|v\|}\right)$	$\cosh(\alpha)x + \frac{\sinh(\alpha)}{\alpha}v,\ \alpha = \sqrt{-K}\|v\|_{\mathbb{L}}$
$\mathrm{Log}_x y$	$\mathrm{PT}_{\mathbf{0}\to x}^{\mathbb{K}}(\mathrm{Log}_{\mathbf{0}}^{\mathbb{K}}((-x)\oplus_{\mathrm{E}} y))$	$\frac{2}{\sqrt{-K}\lambda_x^K}\tanh^{-1}\!\left(\sqrt{-K}\|-x\oplus_{\mathrm{M}} y\|\right)\frac{-x\oplus_{\mathrm{M}} y}{\|-x\oplus_{\mathrm{M}} y\|}$	$\frac{\cosh^{-1}(\beta)}{\sqrt{\beta^2-1}}(y - \beta x),\ \beta = K\langle x, y\rangle_{\mathbb{L}}$
$\mathrm{PT}_{x\to y}(v)$	$d\phi_{\mathbb{P}\to\mathbb{K}} \circ \mathrm{PT}_{\phi(x)\to\phi(y)}^{\mathbb{P}} \circ d\phi_{\mathbb{K}\to\mathbb{P}}(v)$	$\frac{\lambda_x^K}{\lambda_y^K}\mathrm{gyr}[y, -x]v$	$v - \frac{K\langle y, v\rangle_{\mathbb{L}}}{1+K\langle x, y\rangle_{\mathbb{L}}}(x + y)$

The three models are connected by Riemannian isometries given in Eqs. (26) and (27).

$$\phi_{\mathbb{K}\to\mathbb{P}}(x) = \frac{x}{1+\sqrt{1+K\|x\|^2}}, \qquad \phi_{\mathbb{P}\to\mathbb{K}}(y) = \frac{2y}{1-K\|y\|^2}, \tag{26}$$

$$\phi_{\mathbb{K}\to\mathbb{L}}(x) = \left(\frac{\gamma_x}{\sqrt{-K}},\ \gamma_x x\right), \qquad \phi_{\mathbb{L}\to\mathbb{K}}(x_t, x_s) = \frac{x_s}{x_t}, \tag{27}$$

where $\gamma_x = 1/\sqrt{1+K\|x\|^2}$ is the Lorentz factor. These isometries preserve the geodesic distance and induce correspondences between the gyrovector operations, namely Einstein $\oplus_{\mathrm{E}}$ for Klein, Möbius $\oplus_{\mathrm{M}}$ for Poincaré, and Lorentz $\oplus_{\mathbb{L}}$ for the hyperboloid. See Sec. C.4.2.

Notation. For $x \in \mathbb{K}_{\mathrm{K}}^n$ we denote the Lorentz factor by $\gamma_x = 1/\sqrt{1+K\|x\|^2}$. For $x \in \mathbb{P}_K^n$ we denote the conformal factor by $\lambda_x^K = 2/(1+K\|x\|^2)$. For $x \in \mathbb{L}_K^n$ we write $x = (x_t, x_s)$ with $x_t \in \mathbb{R}$ the time coordinate and $x_s \in \mathbb{R}^n$ the spatial component. Throughout we assume $K < 0$, so $\sqrt{-K} \in (0, \infty)$.

Klein closed-form distance. The Klein geodesic distance admits the closed-form $\mathrm{arccosh}$ expression

$$d^{\mathbb{K}}(x, y) = \frac{1}{\sqrt{-K}}\,\mathrm{arccosh}\left(\frac{1+K\langle x, y\rangle}{\sqrt{(1+K\|x\|^2)(1+K\|y\|^2)}}\right), \tag{28}$$

which matches the geodesic distance on $\mathbb{P}_K^n$ and $\mathbb{L}_K^n$ under the isometries (26)–(27).

Klein exp/log maps at a general point. For $p \in \mathbb{K}_{\mathrm{K}}^n$ and $v \in T_p\mathbb{K}_{\mathrm{K}}^n$ with Klein norm $\|v\|_p = \|v\|_{\mathbb{K}}$, the exponential map is (Mao et al., 2024)

$$\mathrm{Exp}_p^{\mathbb{K}}(v) = p + \frac{\sinh(\|v\|_p)\,\hat{v}}{\cosh(\|v\|_p) + \gamma_p^2\langle p, \hat{v}\rangle\sinh(\|v\|_p)}, \qquad \hat{v} = v/\|v\|_p. \tag{29}$$

Since Klein geodesics are straight line segments, $\mathrm{Exp}_p^{\mathbb{K}}(v)$ is a point on the chord through p in direction v, with a nonlinear reparameterization that accounts for the varying Klein metric.

Klein parallel transport. Parallel transport from the origin to $p \in \mathbb{K}_{\mathrm{K}}^n$ admits the closed form (Mao et al., 2024)

$$\mathrm{PT}_{\mathbf{0}\to p}^{\mathbb{K}}(v) = \frac{\langle p, v\rangle(\sqrt{1+K\|p\|^2} - 2)}{1-\sqrt{1+K\|p\|^2}}\,p + \sqrt{1+K\|p\|^2}\,v. \tag{30}$$

At the origin the Klein metric reduces to the Euclidean inner product, so $T_{\mathbf{0}}\mathbb{K}_{\mathrm{K}}^n \cong \mathbb{R}^n$. The general $\mathrm{PT}_{x\to y}^{\mathbb{K}}$ is obtained either from Eq. (30) via left-gyrotranslation or from a Klein–Poincaré conjugation. See Tab. 11.

Proposition C.7 (Gyrovector–Riemannian equivalence (Mao et al., 2024; Ungar, 2022))**.** *The Einstein operations and Riemannian operations on* $\mathbb{K}_{\mathrm{K}}^n$ *are related by*

$$t \otimes_{\mathrm{E}} x = \mathrm{Exp}_{\mathbf{0}}^{\mathbb{K}}\Big(t \cdot \mathrm{Log}_{\mathbf{0}}^{\mathbb{K}}(x)\Big), \tag{31}$$

$$\mathrm{PT}_{\mathbf{0}\to p}^{\mathbb{K}}(v) = \mathrm{Log}_p^{\mathbb{K}}\Big(p \oplus_{\mathrm{E}} \mathrm{Exp}_{\mathbf{0}}^{\mathbb{K}}(v)\Big). \tag{32}$$

These identities let us implement Klein layers using either gyrovector or Riemannian operations interchangeably.

C.4.2 Gyrovector Spaces

Klein space and the Einstein gyrovector space. The Klein space carries the *Einstein gyrovector space* $(\mathbb{K}^n_K, \oplus_{\mathrm{E}}, \otimes_{\mathrm{E}})$ (Ungar, 2022, Ch. 6), with $\oplus_{\mathrm{E}}$ and $\otimes_{\mathrm{E}}$ defined in Eqs. (4) to (5) of the main paper. It satisfies the four gyrogroup axioms (G1–G4) of Def. C.1, where the gyro-identity is $\mathbf{0}$ (so $\mathbf{0} \oplus_{\mathrm{E}} x = x$), the gyro-inverse is the Euclidean negation $\ominus_{\mathrm{E}} x = -x$ (so $(-p) \oplus_{\mathrm{E}} x$ is the centered version of x relative to p), and the Einstein gyration $\mathrm{gyr}_{\mathrm{E}}[x, y]$ takes a closed polynomial form in γ_x, γ_y. Einstein addition obeys the polynomial γ-identity $\gamma_{x \oplus_{\mathrm{E}} y} = \gamma_x \gamma_y (1 - K\langle x, y\rangle)$ and recovers Euclidean addition as $K \to 0^-$. When $K = -1$ and $n = 3$, $\oplus_{\mathrm{E}}$ coincides with the relativistic velocity composition law of special relativity, with $\mathrm{gyr}_{\mathrm{E}}[x, y]$ realizing the Thomas precession (Ungar, 2022).

Poincaré ball and the Möbius gyrovector space. The Poincaré ball carries the *Möbius gyrovector space* $(\mathbb{P}^n_K, \oplus_{\mathrm{M}}, \otimes_{\mathrm{M}})$ (Ungar, 2022, Ch. 6) with gyroaddition

$$x \oplus_{\mathrm{M}} y = \frac{(1 - 2K\langle x, y\rangle - K\|y\|^2)x + (1 + K\|x\|^2)y}{1 - 2K\langle x, y\rangle + K^2\|x\|^2\|y\|^2}, \tag{33}$$

and scalar gyromultiplication coinciding with $\otimes_{\mathrm{E}}$ in coordinates, $t \otimes_{\mathrm{M}} x = \frac{1}{\sqrt{-K}} \tanh\big(t \operatorname{arctanh}(\sqrt{-K}\|x\|)\big) \frac{x}{\|x\|}$. The Möbius gyration formula is given in Tab. 11. The Klein and Poincaré gyrovector spaces are isomorphic via $\phi_{\mathbb{K} \to \mathbb{P}}$ in Eq. (26), so identities transfer between the two.

Lorentz gyrovector space. The Lorentz hyperboloid carries the *Lorentz gyrovector space* $(\mathbb{L}^n_K, \oplus_{\mathbb{L}}, \otimes_{\mathbb{L}})$, derived in Chen et al. (2025b), with gyro-identity $\bar{\mathbf{0}} = (1/\sqrt{-K}, \mathbf{0})$ and gyro-inverse $\ominus_{\mathbb{L}}(x_t, x_s) = (x_t, -x_s)$. The closed-form Lorentz gyroaddition and scalar gyromultiplication are piecewise expressions involving the time and spatial components, given in Chen et al. (2025b) Props. 24–25. We omit the explicit formulas as the Lorentz gyrovector structure is not used by our Klein layers.

Einstein $\oplus_{\mathrm{E}}$ vs. Möbius $\oplus_{\mathrm{M}}$ structural comparison. Although the Klein and Poincaré gyrovector spaces are isomorphic, their addition formulas differ in three ways that make Klein preferable for deep-learning pipelines.

- **Linearity in the second argument.** The numerator of $x \oplus_{\mathrm{E}} y$ is linear in y, while the numerator of $x \oplus_{\mathrm{M}} y$ is quadratic in y and couples $\|y\|^2$ with x. The denominator of $\oplus_{\mathrm{E}}$ is also linear in $\langle x, y\rangle$, whereas $\oplus_{\mathrm{M}}$'s denominator is quartic, containing $\|x\|^2\|y\|^2$. This reduces both the FLOP count and the depth of the autograd graph at every addition site.
- **Polynomial γ-composition law.** Klein γ-values compose polynomially under Einstein addition, $\gamma_{x \oplus_{\mathrm{E}} y} = \gamma_x \gamma_y (1 - K\langle x, y\rangle)$, whereas the Möbius counterpart $\gamma_{x \oplus_{\mathrm{M}} y} = \gamma_x \gamma_y \sqrt{1 - 2K\langle x, y\rangle + K^2\|x\|^2\|y\|^2}$ adds a square root on top of the inputs γ_x, γ_y. The two formulas share the cost of computing γ_x itself (one square root per point), but the Klein composition keeps the autograd graph polynomial at every subsequent Busemann or distance evaluation, whereas the Möbius composition adds one $\sqrt{\cdot}$ node per addition site.
- **Straight geodesics (Beltrami–Klein theorem).** Geodesics of $\mathbb{K}^n_K$ are Euclidean line segments in coordinates, so Klein geodesic hyperplanes coincide with the Euclidean hyperplanes cut by the open ball. Poincaré geodesics are circular arcs, so a geodesic MLR on $\mathbb{P}^n_K$ requires a Möbius action to bend each learned hyperplane.

By contrast, the *scalar multiplication* formulas $t \otimes_{\mathrm{E}} x$ and $t \otimes_{\mathrm{M}} x$ coincide in coordinates because scalar gyromultiplication is intrinsically a rescaling along the unique geodesic through the origin, and that geodesic shares the same $\tanh$ form in both coordinates. The same identity holds for $\mathrm{Exp}_{\mathbf{0}}$ and $\mathrm{Log}_{\mathbf{0}}$. As a consequence, every Klein layer that depends only on origin-based exp/log or scalar multiplication (Klein FC, MLR, BMLR, BFC) inherits its derivation directly from the corresponding Möbius construction by left-gyrotranslation. Only the gyrogroup-aggregation layer (Klein GyroBN, which sums points under $\oplus_{\mathrm{E}}$) is genuinely new, and its closed-form Einstein midpoint has no Möbius analogue.

Summary of Klein operations. Tab. 12 summarizes the Klein operations used in the neural network layers of Sec. 3. Every operation has a closed-form expression and reduces to its Euclidean counterpart as $K \to 0^-$.

Table 12: Klein operations and their roles in neural network layers.

Operation	Formula	Used in
Einstein addition $\oplus_{\mathrm{E}}$	Eq. (4)	BN shift
Einstein scalar mult $\otimes_{\mathrm{E}}$	Eq. (5)	BN rescaling
$\exp_{\mathbf{0}}^{\mathbb{K}}$ / $\log_{\mathbf{0}}^{\mathbb{K}}$	Special case of Eq. (29) at $p{=}\mathbf{0}$	FC tangent lift
Geodesic distance	Eq. (28)	MLR, FC
Busemann function	Eq. (12)	BMLR, BFC
Einstein midpoint	Eq. (22)	BN centering

D Experimental Details

Common protocols.

As we use trivialization tricks in our MLR, FC, and GyroBN layers, all parameters in Klein neural networks lie in Euclidean space. The optimizer is selected per experiment to match the baseline protocol; see each experiment's setup for details. All experiments are conducted on NVIDIA A100 GPUs.

Interpretation of cross-model comparisons.

The Poincaré, Lorentz, and Klein models are isometric realizations of the same hyperbolic space, so coordinate-equivalent constructions have the same representational capacity. We therefore interpret the experiments along three separate axes: predictive performance under the statistical protocol below, computational cost, and the range of settings over which training remains numerically viable.

Significance testing protocol.

For Tabs. 5, 6 and 21, we apply unpaired two-sample Welch t-tests (Welch, 1947) to the reported mean and standard deviation over five seeds. We control multiplicity with the Holm step-down procedure (Holm, 1979), treating the following two questions as separate test families.

Comparison with the column best. In each column, the numerically best result is shown in **bold**, and results that are not significantly worse under the Holm-adjusted Welch test ($p{\geq}0.05$) are underlined. All remaining results are significantly below the column best.

Comparison with the strongest baseline. Each Klein layer is compared with the strongest baseline in the same column. Asterisks indicate a significant positive difference after Holm adjustment: $^{*}p{<}0.05$, $^{**}p{<}0.01$, and $^{***}p{<}0.001$.

D.1 NUMERICAL STABILITY

Gradient norm under feature scaling. The probe of Sec. 4.1 contains no network layer, so anything it measures is attributable to the coordinate map alone. We sample $B{=}256$ unit tangent directions $\hat{v}_j \in \mathbb{R}^{64}$, scale them to $v_j{=}r\hat{v}_j$, map them to $x_j{=}\mathrm{Exp}_{\mathbf{0}}(v_j)$, average $\|x_j\|_2^2$ over the batch and backpropagate that scalar to v_j. Tab. 4 reports the mean norm of the resulting tangent-input gradients in FP32.

The Poincaré and Lorentz models fail through different numerical mechanisms. Lorentz coordinates and their Jacobians scale as $\cosh(\left(\sqrt{-K}r\right))$, allowing gradients to become non-finite even when the forward values remain finite, whereas Poincaré operators lose precision through cancellation near the ball boundary. The FP32–FP64 comparison and the subsequent gradient and round-trip diagnostics distinguish these effects from limitations of floating-point precision alone.

This appendix replicates the stability sweeps of Sec. 4.1 on CIFAR-10 and Tiny-ImageNet, and reports four operator-level diagnostics. The operator-level tests use 256 unit-norm tangent vectors at the origin in $\mathbb{R}^{64}$ with $K{=}{-}1$. Each manifold uses its standard reference implementation as in Sec. 4.1.

CIFAR-10 clipping radius. We rerun the clipping-radius sweep of Sec. 4.1 on CIFAR-10 under the same 100-epoch protocol at fixed $K{=}{-}0.3$. Tab. 13 reports CIFAR-10 top-1 accuracy. Klein trains to a finite accuracy at every radius, while Poincaré and Lorentz NaN at every $r{\geq}5$. The pattern is identical to Tab. 3 on CIFAR-100, confirming that the qualitative separation between manifolds is dataset-agnostic.

Table 13: CIFAR-10 clipping sweep.

r	**Klein**	**Poincaré**	**Lorentz**
0.1	57.37	93.65	67.25
0.5	94.58	94.66	94.42
1.0	94.72	94.55	94.67
2.0	94.59	94.53	94.37
5.0	94.70	**NaN**	**NaN**
10.0	94.24	**NaN**	**NaN**
10^9	93.97	**NaN**	**NaN**

CIFAR-10 curvature. The curvature sweep at fixed $r{=}1.0$ on CIFAR-10 in Tab. 14 reproduces the CIFAR-100 picture, with Klein staying inside a 0.4-point band across all five curvatures, while Poincaré NaNs at $|K|{=}10$ and Lorentz NaNs at $|K|{=}0.01$. The accuracy ceiling on CIFAR-10 is high enough that the within-band variation is below seed noise.

Table 14: CIFAR-10 curvature sweep.

$\lvert K\rvert$	**Klein**	**Poincaré**	**Lorentz**
0.01	94.59	94.74	**NaN**
0.1	94.72	94.82	94.20
0.3	94.69	94.71	94.54
1.0	94.65	94.42	94.46
10.0	94.46	**NaN**	94.45

Tiny-ImageNet clipping radius. The clipping sweep on Tiny-ImageNet, with chance level 0.5%, in Tab. 15 shows the same Klein-vs.-others pattern. Klein is finite across the entire sweep and degrades smoothly when the prior is too aggressive ($r{\leq}0.5$) or too loose ($r{\geq}5$). Poincaré and Lorentz NaN at every $r{\geq}5$ exactly as on CIFAR.

Boundary saturation. We compute $x = \exp_0(r\hat{v})$ for $r \in \{1, 5, 10, 50, 100, 500, 1000\}$ and report the fraction of samples within 10^{-6} of the manifold's projection boundary in Tab. 16. All three manifolds remain finite thanks to their built-in projection. Klein activates its $1{-}10^{-5}$ clamp at $r{=}10$, Poincaré's more conservative $1{-}4 \times 10^{-3}$ threshold never triggers in this range, and the Lorentz time component stays below 10^6.

Round-trip error. For tangent vectors of norm $\|v\| \in \{1, 2, 5, 10, 20, 50\}$ we report the mean error $\|\log_0(\exp_0(v)) - v\|$ in Tab. 17. Klein occupies a middle ground: at $\|v\|{=}5$ in FP32 it is four orders of magnitude more accurate than Poincaré but three orders worse than Lorentz. Switching to FP64

Table 15: Tiny-ImageNet clipping sweep.

r	**Klein**	**Poincaré**	**Lorentz**
0.1	5.98	1.13	6.64
0.5	21.69	1.04	36.92
1.0	61.01	65.43	64.49
2.0	64.87	65.65	65.30
5.0	22.60	**NaN**	**NaN**
10.0	2.34	**NaN**	**NaN**
10^9	0.99	**NaN**	**NaN**

Table 16: Boundary saturation.

Saturation rate (%)	r=1	5	10	50	100	500	1000
Klein	0	0	100	100	100	100	100
Poincaré	0	0	0	0	0	0	0
Lorentz	0	0	0	0	0	0	0

widens Lorentz's lead but barely helps Poincaré, indicating that Poincaré's FP32 error is algorithmic precision loss rather than a representation issue.

Table 17: Log/exp round-trip error.

	FP32						**FP64**					
Manifold	$\|v\|$=1	2	5	10	20	50	$\|v\|$=1	2	5	10	20	50
Klein	4.2e−8	6.6e−7	2.1e−4	3.9	13.9	43.9	1.7e−16	1.7e−15	5.1e−13	3.9	13.9	43.9
Poincaré	2.0e−7	1.6e−6	1.9	6.9	16.9	46.9	1.7e−7	9.3e−7	3.3e−4	3.9	13.9	43.9
Lorentz	**8.9e−8**	**1.0e−7**	**3.0e−7**	**5.9e−7**	**10.0**	**40.0**	**1.3e−16**	**1.8e−16**	**5.7e−16**	**1.3e−15**	**10.0**	**40.0**

Composition drift. We iterate $y_k = \exp_0(\log_0(y_{k-1}))$ with $y_0 = \exp_0(v)$, $\|v\|$=3, and report the relative drift $\|y_k - y_0\|/\|y_0\|$ in Tab. 18. Klein and Lorentz are effectively idempotent under composition: their FP64 drift stays near machine epsilon through k=100. Poincaré drifts linearly in k, reaching 6.1×10^{-6} at k=100 in FP64. Klein and Lorentz are therefore preferable when the manifold operator is restacked across many layers.

Table 18: Composition drift.

	FP32					**FP64**				
Manifold	k=1	5	10	50	100	k=1	5	10	50	100
Klein	5.8e−8	6.7e−8	6.6e−8	6.6e−8	6.6e−8	**1.0e−16**	**1.4e−16**	**1.4e−16**	**1.4e−16**	**1.4e−16**
Poincaré	7.4e−8	2.8e−7	5.4e−7	2.3e−6	4.3e−6	6.1e−8	3.1e−7	6.1e−7	3.1e−6	6.1e−6
Lorentz	1.1e−7	1.4e−7	1.5e−7	1.5e−7	1.5e−7	1.6e−16	2.4e−16	2.4e−16	2.4e−16	2.4e−16

FP32 vs. FP64 training. We re-train each manifold's classification head at the default configuration in FP32 and FP64. As shown in Tab. 19, the accuracy gap is below seed noise for all three manifolds. Training-level precision sensitivity becomes consequential only away from the default radius, which is where the main-text T1 and T2 already record NaN-driven failures.

Table 19: FP32 vs. FP64.

Manifold	FP32	FP64
Klein	77.11	77.10
Poincaré	76.70	76.79
Lorentz	76.98	76.68

Caveats and reproducibility. Each manifold uses its standard projection threshold (Klein $1-10^{-5}$, Poincaré $1-4 \times 10^{-3}$, Lorentz no boundary cap), so the NaN events are intrinsic to these implementations as deployed in the literature. The $K=-0.3$ used in the clipping sweeps follows Guo et al. (2022)'s clipped HNN protocol, and the curvature sweeps span three orders of magnitude under the same pattern, ruling out single-K bias. All cells are single-seed (seed 42).

D.2 Image Classification

D.2.1 Datasets

CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) datasets contain 60K 32×32 colored images from 10 and 100 classes, respectively. We use the dataset split implemented in PyTorch, with 50K training and 10K testing images. Following Bdeir et al. (2024), we use data augmentation that includes random cropping with padding of 4 pixels and random horizontal flipping.

D.2.2 Implementation Details

We implement the experiments using the official code of Bdeir et al. (2024). All models share a ResNet-18 backbone followed by a hyperbolic MLR classifier. The output embedding is mapped to the target manifold via the exponential map at the identity e, that is, $\mathrm{Exp}_e(x)$, with $e = \mathbf{0}$ for the Klein and Poincaré models. The training hyperparameters follow the ResNet-18 protocol of Bdeir et al. (2024) and are listed in Tab. 20. Feature clipping at norm r is applied before $\mathrm{Exp}_{\mathbf{0}}^{\mathbb{K}}$ to prevent overflow near the Klein ball boundary. For Klein heads, the curvature K and clipping radius r are selected per dataset on the grid of Chen et al. (2026a) App. E.1.2 with $K \in \{-1, -0.5, -0.3\}$ and $r \in \{1, 2, 4\}$. The Poincaré and Lorentz baselines retain their original Chen et al. (2026a) settings selected on the same grid, so all methods receive equal hyperparameter budget. We report 5-seed mean$\pm$std on CIFAR-10/100.

Table 20: Training hyperparameters for the ResNet-18 image classification protocol.

Setting	**Value**
Optimizer	SGD
Initial learning rate	0.1
Momentum	0.9
Weight decay	5×10^{-4}
Batch size	128
Epochs	200
LR decay	factor $\gamma = 0.2$ at epochs 60, 120, 160
Curvature	$K \in \{-1, -0.5, -0.3\}$, selected per dataset for Klein heads

D.3 Graph Learning

D.3.1 Datasets

Disease represents a SIR disease transmission tree (2 classes, 1044 nodes, 1043 edges). **Airport** is a transductive dataset of airline routes (4 classes, 3188 nodes, 18631 edges). **PubMed** is a citation network in medicine (3 classes, 19717 nodes, 44338 edges). **Cora** is a citation network in machine learning (7 classes, 2708 nodes, 5429 edges).

D.3.2 Implementation Details

We follow the link-prediction protocol of HGCN (Chami et al., 2019) (data splits, Fermi–Dirac decoder, training and evaluation schedule). Each encoder stacks two hyperbolic FC layers of width 128 with no graph aggregation, followed by a Fermi–Dirac decoder. Klein and PV runs additionally use the GyroBN pipeline of Chen et al. (2025b).

All methods place a hyperbolic ReLU between the two FC layers. Klein FC and PV FC apply σ on the geodesic-distance logit before the $\sinh$ projection back onto the manifold, following the absorbed pattern of Chen et al. (2026b). Klein BFC, Lorentz FC, and PLFC apply σ as a separate module $\mathrm{Log}_{\mathbf{0}} \to \sigma \to \mathrm{Exp}_{\mathbf{0}}$ after each FC layer, denoted KleinHypAct on the Klein ball and LorentzAct on the Lorentz model, following Ganea et al. (2018). Möbius FC, BFC-P, and BFC-L use the in-layer activation of their original papers (Ganea et al., 2018; Shimizu et al., 2021; Chen et al., 2026a). We use $\sigma = \mathrm{ReLU}$ throughout, except Klein BFC on Disease, where $\sigma = \mathrm{SeLU}$ is selected by validation sweep.

All models are trained with the Adam optimizer on an NVIDIA A100 GPU in float32, for up to 5000 epochs with early stopping on the validation ROC. Following HBNN (Chen et al., 2026a) we tune the activation function, learning rate, and dropout per dataset on the validation set, holding the rest of the pipeline fixed. Tab. 22 lists the Klein BFC hyperparameters in the best configuration per dataset

Table 21: Node classification accuracy (%) of FC/BFC heads, 5 seeds. Best results are in **bold**.

Space	Method	Disease (δ=0)	Airport (δ=1)	PubMed (δ=3.5)	Cora (δ=11)
$\mathbb{P}^n_K$	HNN++ (Shimizu et al., 2021)	80.57±0.23	88.40±0.17	73.68±0.39	52.06±0.90
$\mathbb{L}^n_K$	LNN (Bdeir et al., 2024)	79.90±0.01	75.20±1.08	68.82±0.88	53.34±1.65
$\mathbb{L}^n_K$	LNN++ (Shi et al., 2026)	**81.15±0.26**	91.76±1.11	73.52±0.41	52.72±1.45
$\mathbb{PV}^n_K$	PVNN (Chen et al., 2026b)	**81.15±0.23**	97.96±0.42	73.48±0.22	51.42±1.33
$\mathbb{K}^n_K$	KNN (Mao et al., 2024)	79.41±0.55	92.10±0.97	69.36±0.76	52.26±1.99
$\mathbb{K}^n_K$	**KNN++**	80.86±0.34	**98.40±0.83**	74.78±0.74	**54.70±1.20**
$\mathbb{K}^n_K$	**KBNN++**	81.05±0.43	98.06±0.53	**74.84±0.54**	49.04±2.25

(top block) together with the values shared across datasets (bottom block), where r_r denotes the KleinHypAct radius rescaling parameter (default 1.0, with Disease retuned to 0.5). PV FC, PLFC, and Klein FC use the same training protocol with their own per-dataset learning rates and dropout values.

Table 22: Klein BFC link-prediction hyperparameters. Top block lists per-dataset values, bottom block lists values shared across datasets.

Setting	Disease	Airport	PubMed	Cora
σ	SeLU	ReLU	ReLU	ReLU
Learning rate	0.005	0.015	0.005	0.01
Dropout	0.05	0.05	0.05	0.05
Weight decay	10^{-3}	10^{-3}	10^{-3}	0
Patience	200	200	200	300
r_r	0.5	1.0	1.0	1.0
Optimizer	Adam			
Curvature K	-1.0			
Hidden dimension	128			
Number of FC layers	2			
Maximum epochs	5000			
Precision	float32			
Seeds	$\{0, 1, 2, 42, 1234\}$			
Evaluation frequency	every 10 epochs			

D.3.3 NC Backbone Architectures

All compared NC methods share the same template. Input features are lifted to the target manifold via $\mathrm{Exp}_{\mathbf{0}}$, passed through a 2-layer hyperbolic FC encoder (no graph aggregation), and classified by a hyperbolic MLR head. The methods differ in (i) the manifold and (ii) the FC and MLR formulation, summarized below.

HNN++ (Shimizu et al., 2021) uses the Poincaré FC and Poincaré MLR on the Poincaré ball with curvature -1, with Möbius gyro-addition for biases and an in-layer sigmoid activation.

LNN (Bdeir et al., 2024) uses the Lorentz FC encoder and the Lorentz MLR classifier of Bdeir et al. (2024) on the hyperboloid model, with tangent-space activation $\mathrm{Log}_{\mathbf{0}} \to \sigma \to \mathrm{Exp}_{\mathbf{0}}$.

LNN++ (Shi et al., 2026) replaces the LNN encoder with the point-to-hyperplane PLFC layer, which casts each output coordinate as the signed Lorentzian distance to a learnable hyperplane, and retains LMLR as the classifier. We use the official ILNN implementation, swapping only the encoder.

PVNN (Chen et al., 2026b) uses the unbounded PV space $\mathbb{PV}^n_K = \mathbb{R}^n$ with the proper-velocity metric, paired with the PV FC layer and the PV MLR head. We follow the authors' default hyperparameters.

KNN (Mao et al., 2024) is the original Klein NN, where a Klein FC encoder is paired with a Klein MLR head, both formulated through Einstein gyro-operations.

KNN++ is our improved Klein NN, where the closed-form Klein FC encoder of Thm. 3.4 is stacked twice and paired with the Klein MLR head of Thm. 3.2.

KBNN++ keeps the same pipeline but uses the horospheric Klein BFC encoder and the Klein BMLR head, with KleinHypAct activation between FC layers.

D.3.4 PV FC AND PLFC INITIALIZATION

For the PV FC layer (Chen et al., 2026b) and the PLFC layer (Shi et al., 2026) on PubMed, we replace the default LeCun initialization (std 0.02) with Xavier initialization (gain $\sqrt{2}$) for the linear weights. All other defaults are preserved. With this change, PLFC reaches 94.95 ± 0.08 on PubMed, matching the value reported in Tab. 6.

D.3.5 KLEIN NC HYPERPARAMETERS AND TUNING PROTOCOL

For the four Klein cells of Tab. 21 we adopt Chen et al. (2026b)'s feature-only NC pipeline (two-layer Klein FC or Klein BFC backbone), pairing each FC encoder with the matching MLR head, so that KNN++ uses Klein FC with Klein MLR and KBNN++ uses Klein BFC with Klein BMLR. All Klein cells are trained with RIEMANNIANADAM for 2000 epochs with patience 200 under the HGCN splits of Chami et al. (2019). Following Chen et al. (2026b, App. C.3) we reproduce the HNN++, LNN, PVNN, PVNN+GyroBN, and KNN baselines from their original papers. LNN++ (Shi et al., 2026) is not reported on these splits, so we integrate the official PLFC layer into the same pipeline with the LMLR head retained. For LNN++ and the Klein cells the learning rate, the curvature $c = -K$, and the dropout rate are selected by grid search on the validation split at seed 42 and reported as test-split averages over seeds 40 to 44. The full hyperparameter list is in Tab. 23.

Table 23: Klein NC hyperparameters. Top block lists per-dataset values, bottom block lists values shared across datasets.

Setting	**Disease**	**Airport**	**PubMed**	**Cora**
Learning rate	0.01	0.01	0.01	0.005
Curvature $c = -K$	0.3	0.1	1.0	1.0
Dropout	0.1	0.0	0.0	0.0
Optimizer	RIEMANNIANADAM			
Maximum epochs	2000			
Batch size	128			
Hidden dimension	16			
Patience	200 evaluation intervals			
Weight decay	5×10^{-4}			
Tuning seed	42			
Evaluation seeds	$\{40, 41, 42, 43, 44\}$			

D.4 COMPUTATIONAL AND GEOMETRIC ANALYSIS

D.4.1 DECISION BOUNDARIES OF HYPERBOLIC MLR HEADS

Tab. 1 lists what each head computes; Fig. 2 shows what its boundary is. The head is the two-FC-plus-MLR Klein stack of Sec. 4.4 trained on Airport with the last layer set to width two so it can be drawn, reaching 76.7% test accuracy on the 3188 nodes. The top row shows $H = \{x \mid \langle x, z \rangle = r\}$ for one class, the surface whose signed distance is that class logit, with points coloured by the side they fall on. The bottom row shows the partition the same head induces, every pixel classified in Klein coordinates, so the two panels are one classifier seen through two coordinate systems and agree on every point. We draw a class hyperplane rather than a boundary between classes because the latter meet at the origin, where a hyperplane is a diameter in both balls; the one drawn lies 0.555 out, the furthest of the four.

Two independent properties of a boundary drive the cost of a head. It is *totally geodesic* when it is itself a hyperbolic subspace, which is what makes the signed distance to it a geodesic distance; horospheres are not. Separately it may or may not be *flat* in the coordinates the head optimizes over, and only the Klein model gives both at once. The measured bend of H is exactly zero in $\mathbb{K}^n_{\mathrm{K}}$, since Klein geodesics are Euclidean segments (Sec. C.4.2), against 0.232 of the ball radius in $\mathbb{P}^n_K$, where it is a circular arc meeting the boundary orthogonally. In $\mathbb{L}^n_K$ it is the intersection of the hyperboloid with a linear hyperplane through the ambient origin, flat in $\mathbb{R}^{1+n}$ at the cost of carrying the time coordinate. The bottom row shows the same contrast at the level of the partition, whose cells are bounded by straight chords in $\mathbb{K}^n_{\mathrm{K}}$ and by arcs in $\mathbb{P}^n_K$.

A curved boundary must be anchored, so each Poincaré geodesic head carries a manifold-valued $p_k \in \mathbb{P}^n_K$ per class, which is what Tab. 1 records as lost compact parameters and an order of

magnitude in FLOPs. The Klein heads need only the Euclidean pair (z, r) drawn here, so their per-class computation is one batched matrix product, and the geodesic and horospherical Klein heads end up costing the same (Tab. 24).

D.4.2 Computational Cost of Geodesic and Horospherical Heads

Klein MLR and Klein BMLR have nearly identical FLOP counts and runtimes, indicating that the choice between geodesic and horospherical boundaries introduces no material computational difference within the Klein model (Tab. 24). The larger cost of competing heads arises from their manifold-valued class parameters, which require per-class operations instead of a single batched matrix product. Thus, runtime reflects both arithmetic complexity and the batchability of the parameterization.

Table 24: Computational cost of geodesic and horospherical classification heads.

Head	Boundary	Manifold-valued param	FLOPs/class	ms/iter, n=50
Poincaré MLR (Ganea et al., 2018)	geodesic hyperplane	$p_k \in \mathbb{P}_K^n$	$C(19n{+}27)$	—
Pseudo-Busemann MLR (Nguyen et al., 2025)	horosphere	$p_k \in \mathbb{P}_K^n$	$C(18n{+}30)$	217.98
Klein MLR (Sec. 3.1)	geodesic hyperplane	none	$C(2n{+}14)$	1.498
Klein BMLR (Sec. 3.2)	horosphere	none	$C(2n{+}11)$	1.504

D.4.3 Comparison of Geodesic and Horospherical Decoding

Klein FC/BFC and Klein MLR/BMLR provide controlled comparisons in which only the boundary type changes (Tab. 25). Horospherical decoding gives its largest improvement and lower variance on Disease, the exact tree, whereas geodesic decoding performs better on Cora, the least hyperbolic graph. The intermediate datasets do not follow a monotonic trend in δ, indicating that neither boundary type is uniformly preferable and motivating the inclusion of both variants.

Table 25: Geodesic and horospherical decoding within the Klein model. Graph datasets are ordered by δ-hyperbolicity. **Bold** marks the higher value in each row.

Setting	Geodesic head	Horosphere head	Δ
Disease, δ=0	73.10 ± 4.02	82.18 ± 0.39	+9.08
Airport, δ=1	94.54 ± 0.12	95.47 ± 0.36	+0.93
PubMed, δ=3.5	94.69 ± 0.23	95.99 ± 0.13	+1.30
Cora, δ=11	94.09 ± 0.28	92.77 ± 0.43	−1.32
CIFAR-100	78.25 ± 0.13	78.24 ± 0.13	−0.01

D.5 MLR Head Runtime

We benchmark the hyperbolic MLR heads on random Euclidean features at batch size 1024, 100 classes and FP32, averaging a forward+backward pass over five runs after warm-up (Sec. D.2). Tab. 26 reports the mean per-iteration runtime. The runtime ordering follows the FLOP ranking in Tab. 1. PBMLR-P is the slowest due to its high complexity and lack of batch efficiency, whereas the Klein variants consistently achieve the lowest runtime across all input dimensions, demonstrating that the closed-form Klein geometry yields tangible efficiency gains in practice.

Table 26: MLR head runtime (ms / iter).

Method	n=50	n=100	n=200	n=500
Euclidean MLR	0.263	0.273	0.265	0.269
PMLR	2.001	1.999	1.962	2.046
PBMLR-P	217.98	217.86	222.30	221.86
BMLR-P	1.940	1.957	1.932	2.041
LMLR	2.528	2.481	2.512	2.510
BMLR-L	1.696	1.684	1.695	1.801
PV MLR	2.231	2.201	2.206	2.272
Klein MLR	1.498	1.527	1.538	1.502
Klein BMLR	1.504	1.486	1.504	1.551

D.6 Statistical and Robustness Analysis

D.6.1 Significance Tests for Accuracy Comparisons

Tabs. 27 and 28 report the statistical tests underlying the markings in Tabs. 5, 6 and 21, following the protocol in Sec. D. On each image dataset, the better Klein head is compared with both the Euclidean baseline and the strongest hyperbolic baseline; for graph tasks, each Klein model is compared with the strongest baseline in the corresponding column. Statistically significant improvements occur on the graph benchmarks, while none of the reported differences significantly favors a baseline.

Table 27: CIFAR, 5 seeds per group. Δ is Klein minus the comparator, so a positive value favors Klein. Holm correction over the four tests shown.

Comparison	Δ	95% CI	Holm p
Klein MLR − Euclidean, CIFAR-100	$+0.53$	$[+0.32, +0.74]$	0.0014
Klein BMLR − Euclidean, CIFAR-10	$+0.19$	$[+0.04, +0.34]$	0.065
Klein MLR − BMLR-P, CIFAR-100	$+0.15$	$[-0.28, +0.58]$	0.82
Klein BMLR − BMLR-P, CIFAR-10	$+0.01$	$[-0.16, +0.18]$	0.89

Table 28: Graphs, 5 seeds. Klein minus the strongest competing method of that column, for link prediction (LP) and node classification (NC), ordered by δ-hyperbolicity. Stars give the Holm-adjusted significance within each table, $^{*}p < 0.05$, $^{***}p < 0.001$.

Task	Klein layer	Disease (δ=0)	Airport (δ=1)	PubMed (δ=3.5)	Cora (δ=11)
LP	Klein BFC	$+1.73^{*}$	$+0.08$	$+0.70^{***}$	-1.26
NC	KNN++	-0.29	$+0.44$	$+1.10$	$+1.36$
NC	KBNN++	-0.10	$+0.10$	$+1.16^{*}$	-4.30

On Cora link prediction, Klein BFC is 1.26 points below the strongest baseline. This difference is specific to the horospherical decoder: the geodesic Klein FC reaches 94.09 ± 0.28, the highest value in the corresponding column.

D.6.2 Numerical Differences Across Realizations

The three realizations are isometric and therefore have the same representational capacity, while their numerical implementations remain coordinate dependent. Tab. 29 orders the settings by their deviation from the default configuration. When all three models train successfully, the largest absolute difference between Klein and the strongest baseline is 0.19; under more extreme curvature or clipping, the baselines fail to return finite results while Klein remains trainable.

Table 29: CIFAR-100 top-1 (%), ordered by how far the setting is pushed from the default K=−1, r=1. "NaN 3/3" means all three seeds reached a non-finite loss in the first epoch. No margin here is significant.

Setting	Klein	Best baseline	Margin
curvature flattened to $\lvert K\rvert$=0.01	72.51	NaN 3/3	—
clip loosened to r=5	73.98	NaN 3/3	—
each method at its own tuned K and r	78.25	78.10	$+0.15$
default K=−1, r=1 for all three	78.07	78.26	-0.19

D.6.3 Training Viability under Challenging Settings

Table 30: Settings at which a baseline returns no usable model, under the 200-epoch protocol of the manuscript. "NaN" means the training loss became non-finite in the first epoch, "collapse" means the encoder settled at chance.

Setting	Poincaré	Lorentz	Klein
feature radius $r \geq 5$	NaN	NaN	finite at every radius
curvature $\lvert K\rvert$=0.01	NaN	NaN	76.45
BFC encoder at depth 4	collapse on Airport	collapse on both	trains on both
100 composed $\mathrm{Log}_{\mathbf{0}} \circ \mathrm{Exp}_{\mathbf{0}}$	drifts to 6.1×10^{-6}	machine epsilon	machine epsilon
batch center at every step	iterative Fréchet	iterative Fréchet	single $O(mn)$ pass

D.6.4 Effect of Encoder Depth

Tab. 31 evaluates BFC encoders with depths from 2 to 6 on Disease and Airport over five seeds. On Airport, Klein remains trainable through depth 6, whereas Poincaré becomes unstable at depth 2 and both baselines collapse by depth 3; consequently, Klein is the only realization that remains trainable at depth 4 on both datasets. On Disease, Poincaré remains trainable one layer deeper than Klein, while Lorentz collapses from depth 3; at depth 8, all three realizations collapse on both datasets.

Table 31: Link-prediction ROC-AUC (%) against encoder depth, BFC encoder, 5 seeds. "collapse" means AUC at chance, "collapse k/5" that k of the 5 seeds collapsed.

Dataset	Depth	Klein	Poincaré	Lorentz
Disease	2	73.89 ± 0.93	72.46 ± 2.08	74.71 ± 0.68
Disease	3	78.00 ± 0.92	77.20 ± 1.70	collapse
Disease	4	77.65 ± 1.31	77.47 ± 0.83	collapse
Disease	5	collapse 4/5	76.17 ± 0.63	collapse
Disease	6	collapse	collapse 4/5	collapse
Airport	2	93.81 ± 0.36	60.47 ± 14.94 (collapse 3/5)	93.82 ± 0.24
Airport	3	95.14 ± 0.56	collapse 4/5	collapse
Airport	4	94.18 ± 0.26	collapse	collapse
Airport	5	91.47 ± 2.74	collapse	collapse
Airport	6	83.52 ± 4.52	collapse	collapse

D.6.5 Node Classification Compared with the Strongest Baseline

Table 32: Node classification, best Klein head against the strongest competing method in each column, ordered by δ-hyperbolicity. Stars give the Holm-adjusted significance, $^*p < 0.05$; unstarred entries are within noise.

Dataset	δ	Best Klein head	Strongest baseline	Δ
Disease	0	81.05 ± 0.43 (KBNN++)	81.15 ± 0.26 (LNN++)	-0.10
Airport	1	98.40 ± 0.83 (KNN++)	97.96 ± 0.42 (PVNN)	$+0.44$
PubMed	3.5	74.84 ± 0.54 (KBNN++)	73.68 ± 0.39 (HNN++)	$+1.16^*$
Cora	11	54.70 ± 1.20 (KNN++)	53.34 ± 1.65 (LNN)	$+1.36$

Compared with the strongest baseline in each column, the Klein models achieve a significant improvement on PubMed and higher mean accuracy on Airport and Cora, while matching the baseline on Disease.

D.7 Learned Representations

D.7.1 Class Geometry

For each CIFAR-100 class, we compute the mean penultimate feature and measure the angles between all 4950 pairs of class directions. Because these angles are invariant to global rotation and rescaling, they characterize the geometry of the learned representations independently of the coordinate system. The angular spread differs by 0.10° between the two Klein seeds and by the same amount between Klein and Poincaré, while all models have mean angles near 90°; Lorentz exhibits a larger spread of 7.73°.

Table 33: Class geometry on CIFAR-100. Nearest-direction agreement is the fraction of samples in the outer half of the representation whose nearest class direction matches the ground-truth class; chance agreement is 0.01.

Run	Mean angle	Spread of angles	Nearest-direction agreement	Accuracy
Klein, seed 0	90.54°	4.52°	0.928	78.07
Klein, seed 1	90.54°	4.62°	0.921	78.11
Poincaré, seed 0	90.55°	4.42°	0.937	78.26
Lorentz, seed 0	90.46°	7.73°	0.904	77.24

D.7.2 Representation Comparison Across Isometric Formulations

Under matched CIFAR-100 conditions ($K=-1$, $r=1$, and an identical training protocol), we compare the penultimate representations of the Klein, Poincaré, and Lorentz MLR models using linear CKA (Kornblith et al., 2019). Same-model cross-seed CKA provides an empirical reference for the

variation induced by training stochasticity. This analysis tests whether the three isometric formulations learn comparable representations in practice.

Table 34: Representation similarity measured by linear CKA on CIFAR-100 penultimate features at $K=-1$.

Comparison	CKA	Interpretation
Same-model cross-seed	0.86 / 0.86 / 0.89	empirical cross-seed reference
Klein vs. Poincaré	0.86	comparable to the cross-seed reference
Klein vs. Lorentz	0.76	below the cross-seed reference
Poincaré vs. Lorentz	0.75	below the cross-seed reference

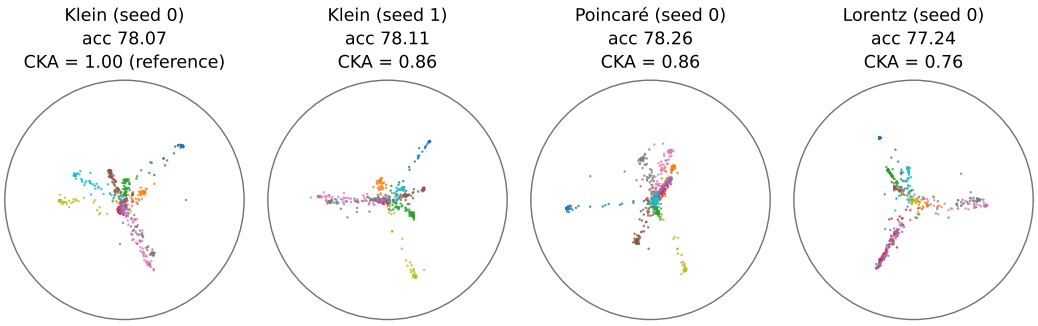


Figure 4: PCA visualizations of the learned representations for the three formulations.

Fig. 4 visualizes the clipped penultimate features of the first ten CIFAR-100 classes at matched $K=-1$. Each panel is reduced independently by PCA and rendered in a Poincaré disk; the visualization therefore supports a qualitative comparison of within-panel class organization rather than direct pointwise alignment across formulations. All three panels exhibit a dense central region with radially separated class clusters.

Klein and Poincaré attain a CKA of 0.86, comparable to the same-model cross-seed reference. Comparisons involving Lorentz yield lower CKA values of 0.75–0.76. Together with their similar classification accuracy, these results indicate substantial representational similarity across the three formulations, while leaving room for coordinate-dependent differences in training dynamics.

E Additional Details of Klein Neural Network Layers

This appendix provides derivations, implementation details, and computational analyses for the five Klein layers introduced in Sec. 3.

Overview of the layer derivations.

The five Klein layers require closed-form expressions for their underlying geometric primitives in Klein coordinates (Tab. 35). Klein MLR follows the signed-margin reformulation of Euclidean softmax and uses the point-to-hyperplane distance established in Thm. 3.1. Klein BMLR replaces the Euclidean inner product with the Klein Busemann function derived in Thm. 3.3, yielding a horospherical classifier directly in Klein coordinates.

Table 35: Closed-form geometric primitives used by the five Klein layers.

Layer	Required primitive
Klein MLR, Klein FC	closed-form signed distance from a point to a Klein hyperplane (Thm. 3.1)
Klein BMLR, Klein BFC	closed-form Klein Busemann function (Thm. 3.3)
Klein GyroBN	closed-form Einstein midpoint for single-pass batch centering

E.1 Klein MLR and BMLR

Klein Busemann derivations.

Derivation via Klein–Lorentz isometry. The Lorentz Busemann function in direction $v \in \mathbb{S}^{n-1}$ is $B^v_{\mathbb{L}}(x) = \frac{1}{\sqrt{-K}} \log(\sqrt{-K}(x_t - \langle x_s, v\rangle))$ (Chen et al., 2026a). The Klein–Lorentz isometry Eq. (27) lifts $x_K \in \mathbb{K}^n_K$ to $(x_t, x_s) = (\gamma_{x_K}/\sqrt{-K}, \gamma_{x_K} x_K)$ with $\gamma_{x_K} = 1/\sqrt{1 + K\|x_K\|^2}$. Substituting,

$x_t - \langle x_s, v\rangle = \gamma_{x_K}(1/\sqrt{-K} - \langle x_K, v\rangle)$, so

$$B^v_{\mathbb{K}}(x_K) = B^v_{\mathbb{L}}(\phi_{\mathbb{K}\to\mathbb{L}}(x_K)) = \frac{1}{\sqrt{-K}}\log\left(\frac{1-\sqrt{-K}\langle x_K, v\rangle}{\sqrt{1+K\|x_K\|^2}}\right),$$

which agrees with Eq. (12).

Cross-check via Klein–Poincaré isometry. We cross-check Eq. (12) against the closed-form Poincaré Busemann function (24) listed in Sec. C.2. The Klein–Poincaré isometry Eq. (26) maps $x_K \in \mathbb{K}^n_{\mathrm{K}}$ to $x_P = x_K/(1+\sigma)$ with $\sigma = \sqrt{1+K\|x_K\|^2}$. We evaluate Eq. (24) at x_P in four steps.

Step 1 ($\|x_P\|^2$ and $1+K\|x_P\|^2$).

$$\|x_P\|^2 = \frac{\|x_K\|^2}{(1+\sigma)^2},$$

$$1+K\|x_P\|^2 = \frac{(1+\sigma)^2 + K\|x_K\|^2}{(1+\sigma)^2} = \frac{(1+\sigma)^2+\sigma^2-1}{(1+\sigma)^2} = \frac{2\sigma(1+\sigma)}{(1+\sigma)^2} = \frac{2\sigma}{1+\sigma},$$

using $\sigma^2 = 1+K\|x_K\|^2$.

Step 2 ($\|v - \sqrt{-K}\,x_P\|^2$).

$$\begin{aligned}\|v-\sqrt{-K}\,x_P\|^2 &= 1 - \frac{2\sqrt{-K}\langle x_K, v\rangle}{1+\sigma} + \frac{(-K)\|x_K\|^2}{(1+\sigma)^2}\\ &= 1 - \frac{2\sqrt{-K}\langle x_K, v\rangle}{1+\sigma} + \frac{1-\sigma^2}{(1+\sigma)^2} = \frac{2(1-\sqrt{-K}\langle x_K, v\rangle)}{1+\sigma}.\end{aligned}$$

Step 3 (ratio).

$$\frac{\|v-\sqrt{-K}\,x_P\|^2}{1+K\|x_P\|^2} = \frac{2(1-\sqrt{-K}\langle x_K, v\rangle)/(1+\sigma)}{2\sigma/(1+\sigma)} = \frac{1-\sqrt{-K}\langle x_K, v\rangle}{\sqrt{1+K\|x_K\|^2}},$$

yielding

$$B^v_{\mathbb{P}}(x_P) = \frac{1}{\sqrt{-K}}\log\left(\frac{1-\sqrt{-K}\langle x_K, v\rangle}{\sqrt{1+K\|x_K\|^2}}\right). \tag{34}$$

Step 4 (consistency check). Both the Lorentz route and the Poincaré route yield the same expression (34), which is Eq. (12). The two derivations are therefore fully consistent.

Remark E.1 (Consistency of Lorentz and Poincaré derivations). Eq. (12) (Lorentz route) and Eq. (34) (Poincaré route) produce identical expressions. The two independent derivations therefore confirm Eq. (12).

Euclidean limit. As $K \to 0^-$, writing $x_K = \tilde{x}/R$ with $R = 1/\sqrt{-K} \to \infty$ gives $\langle x_K, v\rangle \to 0$ and $K\|x_K\|^2 \to 0$, so $B^v_{\mathbb{K}}(x_K) \approx R\cdot(-\langle \tilde{x}, v\rangle/R) = -\langle \tilde{x}, v\rangle$, which is the Euclidean Busemann $B^v_{\mathrm{Euc}}(\tilde{x})$.

Equivalence of the unidirectional MLR parameterization.

The raw two-vector and unidirectional forms parameterize the same family of Klein geodesic hyperplanes. Setting $v_k(x) = 0$ in Eq. (10) gives $\langle x, \hat{z}_k\rangle = \tanh(\sqrt{-K}r_k)/\sqrt{-K}$, the Hesse normal form with unit normal $\hat{z}_k$ and offset r_k, so every Klein geodesic hyperplane is represented by some $(\hat{z}_k, r_k)$. Conversely, expanding $\langle(-p_k)\oplus_{\mathrm{E}} x, w_k\rangle = 0$ recovers $\hat{z}_k$ and r_k from a trained raw head. Converted checkpoints reproduce the raw classifier within 2.6×10^{-11} in float64 with 100% argmax agreement, and Tab. 36 provides a controlled comparison at both standard and low embedding dimensions.

The unidirectional parameterization reduces the number of parameters per class from $2n$ to $n+1$, with all parameters remaining in Euclidean space. More importantly, its class dependence enters through $\langle x, z_k\rangle$, allowing the C logits of a batch X to be evaluated by a single matrix product XZ. In the raw two-vector form, the class parameter appears inside the gyroaddition, preventing this factorization and requiring either a class-wise loop or a $[\mathrm{bs}, C, n]$ broadcast tensor. Tab. 37 reports the resulting head-level runtime and memory costs.

Comparison criteria.

Table 36: Comparison of unidirectional and raw two-vector Klein MLR parameterizations under matched training conditions, with five seeds per setting.

Setting	Unidirectional (n+1)	Raw two-vector ($2n$)
CIFAR-10	95.29 ± 0.09	95.25 ± 0.11
CIFAR-100	78.19 ± 0.16	78.11 ± 0.21
CIFAR-10, d=2	78.94 ± 0.60	78.62 ± 0.83
CIFAR-10, d=4	91.28 ± 0.24	91.19 ± 0.30
CIFAR-10, d=10	94.36 ± 0.13	94.31 ± 0.16

Table 37: Head-level forward and backward cost at batch size 128 on an NVIDIA A100 64GB. Runtime is reported relative to the unidirectional form; OOM denotes out of memory.

(n, C)	Raw two-vector runtime	Unidirectional peak memory
$(512, 10)$	$3.3\times$	—
$(4096, 1000)$	$157\times$	—
$(4096, 10000)$	OOM	0.8 GiB

Tab. 1 compares hyperbolic MLR methods along the underlying space, the logit form, the type of point-to-hyperplane distance (Dist), parameter compactness, FLOPs, and batch efficiency. We expand on the three columns that warrant additional explanation, mirroring the discussion of Chen et al. (2026a).

1. **Hyperplanes.** Klein MLR uses geodesic hyperplanes of the Klein ball, defined as the level sets $\{x \in \mathbb{K}^n_{\mathrm{K}} : \langle \mathrm{Log}^{\mathbb{K}}_p(x), a\rangle_p = 0\}$, while Klein BMLR uses Busemann-based horospheres, the level sets $\{x \in \mathbb{K}^n_{\mathrm{K}} : B^v_{\mathbb{K}}(x) = -b/\alpha\}$. Both Klein variants admit (i) compact parameterization that avoids per-class manifold-valued points, (ii) a natural intrinsic generalization of Euclidean hyperplanes (geodesic) and inner-product separators (Busemann), and (iii) closed-form distances. In contrast, Lorentz MLR (Bdeir et al., 2024) relies on the ambient Minkowski space and is tailored to the Lorentz model, while the Poincaré-MLR baselines (Ganea et al., 2018; Nguyen et al., 2025) are over-parameterized because each class carries a manifold-valued base point $p_k \in \mathbb{P}^n_K$.
2. **Point-to-hyperplane distances (Dist).** **Real** indicates the geodesic point-to-hyperplane distance $\inf_{q\in H} d(x, q)$, where H is a hyperbolic hyperplane and d is the geodesic distance on the manifold; this is the natural Riemannian generalization of the Euclidean signed margin. **Pseudo** indicates a surrogate distance that coincides with the real distance only in the Euclidean limit. The Pseudo-Busemann MLR (Nguyen et al., 2025) uses such a surrogate, while all Klein variants (Klein MLR via Thm. 3.1, Klein BMLR via the closed-form Klein Busemann of Thm. 3.3) compute the real distance, ensuring geometric fidelity across the underlying manifold.
3. **Batch efficiency.** Several Poincaré variants (Ganea et al., 2018; Nguyen et al., 2025) compute logits using $\langle -p_k \oplus_{\mathrm{M}} x,\ a_k\rangle$ and $\| -p_k \oplus_{\mathrm{M}} x\|^2$. For a batch $X \in \mathbb{R}^{\mathrm{bs}\times n}$ and C classes, evaluating $-p_k \oplus_{\mathrm{M}} X$ for every k yields an intermediate tensor of shape $[\mathrm{bs}, C, n]$ that can cause GPU out-of-memory (OOM) when n or C is large. Their official implementations therefore loop over classes, which is batch-inefficient. By contrast, Klein BMLR's logit $-\alpha_k B^{v_k}_{\mathbb{K}}(x) + b_k$ reduces to the class-wise inner product $\langle v_k, x\rangle$ on top of per-input shared quantities (Eq. (13)). Stacking $V = [v_1, \ldots, v_C] \in \mathbb{R}^{n\times C}$, this is implementable as a single matrix multiplication XV, with no $[\mathrm{bs}, C, n]$ intermediate; Klein MLR achieves the same matmul-batched form via the unidirectional parameterization of Thm. 3.2.

FLOP comparison.

The side-by-side comparison of hyperbolic MLR methods appears in main paper Tab. 1. All FLOPs are reported under the *unified amortized-strict convention*, where per-sample shared quantities such as $\|x\|^2$ and γ_x are amortized across the C classes within a forward pass and elementary operations are counted strictly following (Chen et al., 2026a) with no fused multiply-add. Some entries differ from the original publications, which used a non-amortized counting style. We have re-derived them under our unified convention for direct comparison, with the relative ordering of methods preserved (see Sec. E.1.1 for per-method derivations and the cross-walk between conventions). Among methods with unconstrained parameters, Klein BMLR achieves the unique fastest count $C(2n{+}11)$, strictly

beating Lorentz BMLR ($C(2n{+}12)$) and the unit-normalized BMLR-P ($C(2n{+}12)$). Klein MLR is within 3 FLOPs of Lorentz BMLR at $C(2n{+}14)$, and all four bounded-ball Busemann variants beat every Poincaré-geodesic MLR by a factor of 2–10.

E.1.1 FLOP Counting Convention

The FLOP values in Tab. 1 are reported under a single *amortized-strict* convention. Per-sample shared quantities such as $\|x\|^2$, $\gamma_x{=}1/\sqrt{1{+}K\|x\|^2}$ are amortized across the C classes within each forward pass, while elementary operations are counted strictly so that all methods sit on a common axis. Several values therefore differ from the numbers given in the corresponding original publications, while the relative ordering of methods is unchanged.

The convention.

- **Amortization.** Per-sample shared quantities (e.g. $\|x\|^2$, $\gamma_x{=}1/\sqrt{1{+}K\|x\|^2}$, $\sqrt{-K}\,x$) are computed once per forward pass and reused across all C classes, so their per-class cost is 0.
- **Strict elementary operations.** We follow Chen et al. (2026a), where $\log$ and $\exp$ count 6 FLOPs each, $\sinh$ and $\cosh$ count 14 each, $\operatorname{arcsinh}$ and atanh count 10 each, and $\sqrt{\cdot}$ counts 1. An n-dimensional inner product counts $2n{-}1$ FLOPs (no fused multiply-add).
- **Per-iteration precomputation.** Class-only quantities that depend on the class parameters but not on the input batch are computed once per training iteration and amortized over the forward pass. Examples include $\cosh(\sqrt{-K}r_k)$, $\sinh(\sqrt{-K}r_k)$, $\|z_k\|$ for Klein MLR, and $\sqrt{-K}b_k$, $\cosh(\cdot)$, $\sinh(\cdot)$ for Shimizu's MLR. Their per-class cost in Tab. 1 is 0.

Difference from non-amortized literature counts. Several baselines were originally reported without per-sample amortization, which inflates each method's per-class cost by roughly $2n{+}2$ to $2n{+}5$ FLOPs (one $\|x\|^2$, one $K\|x\|^2$, one $1{+}K\|x\|^2$, plus a possible $\sqrt{-K}\,x$ vector reconstruction). For Ganea's Poincaré MLR (Ganea et al., 2018) this raises $C(19n{+}27)$ to the original $C(19n{+}29)$. For Shimizu's Poincaré MLR (Shimizu et al., 2021) it raises $C(2n{+}46)$ to the original $C(4n{+}52)$. For the Bdeir Lorentz MLR (Bdeir et al., 2024) it raises $C(4n{+}51)$ to the original $C(4n{+}52)$. The Pseudo-Busemann MLR (Nguyen et al., 2025) is reported here as $C(18n{+}30)$ rather than the original $C(19n{+}34)$ for the same reason plus a small constant rounding. None of these differences alters the relative ordering of methods, and our claim that *all bounded-ball Busemann variants beat every Poincaré-geodesic MLR by a factor of* 2–10 is conservative because under the original non-amortized counts the gap is even larger.

The two BMLR-P entries. The original HBNN derivation (Chen et al., 2026a) expresses $\|v_k - \sqrt{-K}\,x\|^2$ by first constructing the difference vector and then taking its norm, costing $4n{-}1$ FLOPs. Under our amortized-strict convention the corresponding per-class cost is $C(4n{+}10)$. The official HBNN code release and our own reproductions, however, use the algebraic expansion $\|v_k - \sqrt{-K}x\|^2 = \|v_k\|^2 - 2\sqrt{-K}\langle v_k, x\rangle + (-K)\|x\|^2$ together with the unit-normalization $\|v_k\| = 1$, reducing the per-class cost to a single inner product plus constants, $C(2n{+}12)$. We report both numbers because the choice is a parameterization decision rather than an algorithmic improvement, and in either case BMLR-P does not beat Klein BMLR's $C(2n{+}11)$.

Structural origin of the Klein BMLR cost. Among hyperbolic MLRs with unconstrained parameters, Klein BMLR attains the lowest count $C(2n{+}11)$. Its logit requires one inner product $\sqrt{-K}\langle x, v_k\rangle$ ($2n$ FLOPs with the pre-multiplied $\sqrt{-K}$) and one $\log$ on the resulting quotient (6 FLOPs), with the denominator $\sqrt{1 + K\|x\|^2}$ reused from the amortized γ_x at 0 per-class cost. The Lorentz Busemann

$$B_{\mathbb{L}}^{v_k}(x) \;=\; \frac{1}{\sqrt{-K}}\,\log\!\big(\sqrt{-K}\,(x_t - \langle x_s, v_k\rangle)\big)$$

applies an additional overall $\sqrt{-K}$ scaling after the subtraction, reaching $C(2n{+}12)$. The Poincaré BMLR-P with unit-normalized v_k also reaches $C(2n{+}12)$ via algebraic expansion. Klein BMLR therefore holds a 1-FLOP-per-class advantage over both Busemann competitors.

Klein MLR vs. Klein BMLR, a 3-FLOP gap within Klein. Within Klein, the geodesic MLR's $C(2n{+}14)$ exceeds the Busemann variant's $C(2n{+}11)$ by 3 FLOPs. The geodesic logit uses an $\operatorname{arcsinh}$ (10 FLOPs), whereas the Busemann logit uses a $\log$ (6 FLOPs) plus the $\sqrt{-K}$ inner-product

Table 38: Comparison of hyperbolic FC layers $\mathcal{F} : \mathbb{H}^n_K \to \mathbb{H}^m_K$. Slowest FLOPs and largest #Params in red, fastest FLOPs and smallest #Params in green.

Method	$\mathcal{F} : \mathbb{H}^n_K \ni x \mapsto y \in \mathbb{H}^m_K$	Space	Methodology	Parameters	#Params	FLOPs
Möbius FC (Ganea et al., 2018)	$y = \frac{1}{\sqrt{-K}} \tanh(\theta) \frac{Wx}{\|Wx\|}$, $\theta = \frac{\|Wx\|}{\|x\|} \tanh^{-1}(\sqrt{-K}\|x\|)$	$\mathbb{P}^n_K$	Tangent	$W \in \mathbb{R}^{m \times n}$	mn	$2nm+2n+2m+24$
Poincaré FC (Shimizu et al., 2021)	$y = \frac{\omega}{1+\sqrt{1-K\|\omega\|^2}}$, $\omega_k = \frac{\sinh(\sqrt{-K}\, u_k(x))}{\sqrt{-K}}$; $u_k(x)$ from Tab. 1 (Shimizu MLR)	$\mathbb{P}^n_K$	Poincaré geodesic	$\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1},\ b_k \in \mathbb{R},\ k \le m$	$m(n+2)$	$4nm+71m+4$
Lorentz FC (Bdeir et al., 2024)	$y = \begin{bmatrix} \sqrt{\|\psi(Wx,v)\|^2 - 1/K} \\ \psi(Wx,v) \end{bmatrix}$, $\psi(Wx,v) = \lambda\, \sigma(v^\top x + b')\, \eta$, $\eta = \frac{W\phi(x)+b}{\|W\phi(x)+b\|}$; ϕ activation, σ sigmoid	$\mathbb{L}^n_K$	Ambient Minkowski	$W \in \mathbb{R}^{m \times (n+1)},\ v \in \mathbb{R}^{n+1},\ b \in \mathbb{R}^m,\ b' \in \mathbb{R},\ \lambda > 0$	$m(n+1)+m$ $+(n+1)+2$	$2nm+8m+2n+10$
PLFC (Shi et al., 2026)	$y_s = \frac{1}{\sqrt{-K}} \sinh(\sqrt{-K}\, v_k(x))$, $y_t = \sqrt{-1/K + \|y_s\|^2}$; $v_k(x)$ from Tab. 1 (Lorentz MLR)	$\mathbb{L}^n_K$	Lorentz geodesic	$z_k \in \mathbb{R}^n,\ a_k \in \mathbb{R},\ b_k \in \mathbb{R},\ k \le m$	$m(n+2)$	$4nm+69m+2$
BFC-P (Chen et al., 2026a)	$y = \frac{\omega}{1+\sqrt{1-K\|\omega\|^2}}$, $\omega = \frac{\sinh(\sqrt{-K}\, u(x))}{\sqrt{-K}}$; $u_k(x) = -\alpha_k B^{v_k}_{\mathbb{P}}(x) + b_k$	$\mathbb{P}^n_K$	Busemann	$\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1},\ b_k \in \mathbb{R},\ k \le m$	$m(n+2)$	$6nm+29m+4$
BFC-L (Chen et al., 2026a)	$y_s = \frac{1}{\sqrt{-K}} \sinh(\sqrt{-K}\, u(x))$, $y_t = \sqrt{-1/K + \|y_s\|^2}$; $u_k(x) = -\alpha_k B^{v_k}_{\mathbb{L}}(x) + b_k$	$\mathbb{L}^n_K$	Busemann	$\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1},\ b_k \in \mathbb{R},\ k \le m$	$m(n+2)$	$2nm+30m+2$
Klein FC Sec. 3.3	$y_k = \frac{\sinh(\sqrt{-K}\, v_k(x))}{\sqrt{-K}\,\rho}$, $\rho = \sqrt{1+\sum_j \sinh^2(\sqrt{-K}\, v_j(x))}$; $v_k(x)$ from Tab. 1 (Klein MLR, Eq. (10))	$\mathbb{K}^n_K$	Klein geodesic	$z_k \in \mathbb{R}^n,\ r_k \in \mathbb{R},\ k \le m$	$m(n+1)$	$2nm+32m+2$
Klein BFC Sec. 3.3	$y_k = \frac{\sinh(\sqrt{-K}\, u_k(x))}{\sqrt{-K}\,\rho}$, $\rho = \sqrt{1+\sum_j \sinh^2(\sqrt{-K}\, u_j(x))}$; $u_k(x) = -\alpha_k B^{v_k}_{\mathbb{K}}(x) + b_k$ (Eq. (13))	$\mathbb{K}^n_K$	Busemann	$\alpha_k > 0,\ v_k \in \mathbb{S}^{n-1}$ $b_k \in \mathbb{R},\ k \le m$	$m(n+2)$	$2nm+29m+2$

scale (1 FLOP), a net saving of 3. This 3-FLOP gap is the only counter-cost of the geodesic-to-Busemann substitution and is more than offset by the accuracy gains in Sec. 4.2.

E.2 Klein FC and BFC

Cost comparison.

Tab. 38 gives the side-by-side comparison of hyperbolic FC layers, extending Chen et al. (2026a, Tab. 3) with our Klein FC and Klein BFC. Each per-output logit is read directly off Tab. 1 and the sinh-normalized Klein output map adds a shared $18m + 2$ FLOPs, as derived step-by-step in Sec. E.2.1. Klein BFC attains the lowest count, $2nm + 29m + 2$, m FLOPs below Lorentz BFC-L ($2nm + 30m + 2$). The saving is the same Klein-quotient 1-FLOP gap that already appears for the MLR layers, accumulated across the m output dimensions. To match the conditions of Chen et al. (2026a, Tab. 3), BFC layers in the table are written without the gyroaddition bias and with the activation ϕ taken as the identity, in line with the Möbius and Lorentz FC entries. FLOPs are counted per single input under the unified amortized-strict convention of Sec. E.1.1.

E.2.1 Klein FC and BFC FLOP Derivation

We derive the Klein FC and Klein BFC entries of Tab. 38 explicitly, using the unified amortized-strict convention of Sec. E.1.1, which counts log and exp at 6 FLOPs each, sinh and cosh at 14, arcsinh and

atanh at 10, $\sqrt{\cdot}$ at 1, and an n-dimensional inner product at $2n-1$. Per-sample shared quantities such as $\|x\|^2$ and γ_x are amortized across the m output dimensions, and per-iteration precomputations that depend only on parameters ($\sqrt{-K}$, $\|z_k\|$, $\cosh(\sqrt{-K}r_k)$, $\sinh(\sqrt{-K}r_k)$, $\alpha_k/\sqrt{-K}$) are computed once per training step. The total decomposes into three additive components.

$$\text{FLOPs}(\mathcal{F}) \;=\; \underbrace{T_{\text{shared}}(x)}_{\text{per-input, }\le\ 2n+2} \;+\; \underbrace{m\cdot T_{\text{logit}}}_{\text{per-output logit}} \;+\; \underbrace{T_{\text{out}}(m)}_{\text{sinh-norm output map}} \quad .$$

Per-input shared cost $T_{\text{shared}}(x)$. Both Klein FC and Klein BFC require the per-input quantity $\gamma_x{=}1/\sqrt{1+K\|x\|^2}$, used inside $v_k(x)$ for FC and inside the denominator of $u_k(x)$ for BFC.

Step	Operation	FLOPs
1	$\|x\|^2$ (n-d squared norm)	$2n{-}1$
2	$K\cdot\|x\|^2$	1
3	$1+K\|x\|^2$	1
4	$\gamma_x{=}\big(1+K\|x\|^2\big)^{-1/2}$ (rsqrt)	1
Total $T_{\text{shared}}(x)$		$\mathbf{2n+2}$

This cost is independent of m, and with m amortization it is $O(1/m)$ per output and does not appear in the leading-order $2nm$ term. We retain it explicitly in the totals below.

Per-output logit cost T_{logit}. The per-output logit $s_k(x)$ is identical to a single MLR class score, so T_{logit} is read off Tab. 1 assuming T_{shared} already done.

- **Klein FC.** $s_k{=}v_k$, see Eq. (10), $T_{\text{logit}}{=}2n{+}14$, hence $m{\cdot}T_{\text{logit}}{=}2nm{+}14m$.
- **Klein BFC.** $s_k{=}u_k$, see Eq. (13), $T_{\text{logit}}{=}2n{+}11$, hence $m{\cdot}T_{\text{logit}}{=}2nm{+}11m$.

Output-map cost $T_{\text{out}}(m)$. The map

$$y_k \;=\; \frac{\sinh(\sqrt{-K}\,s_k)}{\sqrt{-K}\sqrt{1+\sum_{j=1}^{m}\sinh^2(\sqrt{-K}\,s_j)}}, \qquad k{=}1,\dots,m,$$

is identical for Klein FC and Klein BFC. Counted strictly,

Step	Operation	m-term	const
1	$\sqrt{-K}\cdot s_k,\ k{=}1,\dots,m$ ($\sqrt{-K}$ precomputed)	m	0
2	$\sinh(\cdot)$ (14 FLOPs each, m outputs)	$14m$	0
3	$\sinh^2$ (one square per output)	m	0
4	$\sum_{j=1}^{m}\sinh^2$ ($m{-}1$ scalar adds)	$m{-}1$	0
5	$1+\sum$	0	1
6	$\sqrt{1+\sum}$	0	1
7	$\sqrt{-K}\cdot$ (step 6) (denominator factor)	0	1
8	$\sinh(\cdot)$/denom (m scalar divisions)	m	0
Total $T_{\text{out}}(m)$		$\mathbf{18m}$	**2**

Totals. Adding the three components,

$$\text{Klein FC} = \underbrace{(2n{+}2)}_{T_{\text{shared}}} + \underbrace{m(2n{+}14)}_{m\cdot T_{\text{logit}}} + \underbrace{(18m{+}2)}_{T_{\text{out}}} = 2nm+32m+2n+4 \;=\; \mathbf{2nm+32m+O(n)},$$

$$\text{Klein BFC} = \underbrace{(2n{+}2)}_{T_{\text{shared}}} + \underbrace{m(2n{+}11)}_{m\cdot T_{\text{logit}}} + \underbrace{(18m{+}2)}_{T_{\text{out}}} = 2nm+29m+2n+4 \;=\; \mathbf{2nm+29m+O(n)}.$$

The $\mathcal{O}(n)$ tail amortizes to zero per output. The entries reported in Tab. 38 ($2nm{+}32m{+}2$ and $2nm{+}29m{+}2$) follow the same T_{shared}-omitting accounting as the BFC-P and BFC-L entries of Chen et al. (2026a, Tab. 3), which absorb their analogous shared cost into the trailing constant.

Cross-check against Chen et al. (2026a, Tab. 3) BFC-L. The same decomposition applied to Lorentz BFC has $T_{\text{shared}}{=}0$ (BMLR-L uses x_t and x_s directly, with no $\|x\|^2$ shared) and $T_{\text{logit}}{=}2n{+}12$; see Tab. 1. The Lorentz output map $y_s{=}\sinh(\sqrt{-K}u)/\sqrt{-K}$, $y_t{=}\sqrt{-1/K+\|y_s\|^2}$ has cost

$$\underbrace{m}_{\sqrt{-K}\,u_k} + \underbrace{14m}_{\sinh} + \underbrace{m}_{/\sqrt{-K}} + \underbrace{(2m{-}1)}_{\|y_s\|^2} + \underbrace{1}_{-1/K+\cdot} + \underbrace{1}_{\sqrt{\cdot}} \;=\; 18m+1.$$

Total BFC-L is $0{+}m(2n{+}12){+}(18m{+}1) = 2nm{+}30m{+}1$, matching Chen et al. (2026a, Tab. 3)'s reported $2nm{+}30m{+}2$ within a single rounding constant.

PLFC, Lorentz geodesic FC FLOPs. The PLFC layer of Shi et al. (2026) reuses the Lorentz output map of BFC-L but replaces the Busemann logit with the Lorentz geodesic MLR signed margin $v_k(x) = \frac{1}{\sqrt{-K}}\operatorname{sign}(\alpha_k)\beta_k|\operatorname{arcsinh}(\sqrt{-K}\alpha_k/\beta_k)|$ (Lorentz MLR row of Tab. 1, with $T_{\text{logit}} = 4n{+}51$ in the strict-amortized convention, counting $\langle z_k, x_s\rangle$ as $2n$, $\|z_k\|^2$ as $2n$, $\sqrt{-K}a_k$, $\cosh$, $\sinh$ as $1{+}28$, the β_k scalar chain as ~ 8, $\operatorname{arcsinh}$ as 10, and the four trailing scalar combinators as 4). With the same $T_{\text{shared}} = 0$ as BFC-L (Lorentz coordinates x_t and x_s enter directly without $\|x\|^2$ amortization) and the identical Lorentz output map $T_{\text{out}} = 18m{+}1$ derived above,

$$\text{PLFC} = 0 + m(4n{+}51) + (18m{+}1) = 4nm + 69m + 1,$$

reported as $4nm{+}69m{+}2$ in Tab. 38 after absorbing one rounding constant, paralleling the $2nm{+}30m{+}2$ formatting of BFC-L. The PLFC$-$BFC-L gap is

$$\big(4nm{+}69m{+}2\big) - \big(2nm{+}30m{+}2\big) \;=\; 2nm + 39m,$$

i.e. an additional $2n + 39$ FLOPs per output. The $2n$ comes from the extra $\|z_k\|^2$ inner product in the Lorentz MLR margin (Busemann uses only $\langle x_s, v_k\rangle$), and the 39 scalar FLOPs account for the $\cosh$, $\sinh$, and $\operatorname{arcsinh}$ chain replacing Busemann's single $\log$.

Klein BFC vs. BFC-L structural saving. Subtracting the two totals,

$$\text{Klein BFC}-\text{BFC-L} \;=\; [m(2n{+}11){+}18m{+}2{+}2n{+}2] - [m(2n{+}12){+}18m{+}1] \;=\; -m{+}2n{+}3,$$

which is negative whenever $m{>}2n + 3$, that is, for any hidden layer wider than twice the input dimension. The structural 1-FLOP-per-output saving from the Klein quotient form, see Eq. (13), accumulates linearly in m and dominates the small $\mathcal{O}(n)$ shared overhead of computing γ_x.

Klein FC vs. Klein BFC, a $3m$-FLOP gap within Klein. Klein FC $-$ Klein BFC $= 3m$ FLOPs. The gap is exactly 3 per output, the same difference between Klein MLR ($C(2n{+}14)$) and Klein BMLR ($C(2n{+}11)$) in Tab. 1: each Klein FC output uses an $\operatorname{arcsinh}$ (10 FLOPs), whereas each Klein BFC output uses a $\log$ (6 FLOPs) plus a $\sqrt{-K}$ inner-product scale (1 FLOP), a net saving of 3. As with the MLR comparison, this $3m$-FLOP cost is more than offset by the accuracy gains in Sec. 4.2.

Activation absorption. The normalization $\sqrt{1+\sum_j \sinh^2(\sqrt{-K}\,s_j)}$ in Eq. (15) plays the role of the softmax denominator on the Klein ball, enforcing $\|y\|^2 < -1/K$, while the unbounded PV layer (Chen et al., 2026b) omits this factor. An activation $\phi\colon \mathbb{R}\to\mathbb{R}$ can be absorbed by replacing $s_k(x)$ with $\phi(s_k(x))$ in Eq. (15). Since the score space is an unconstrained $\mathbb{R}$, this is numerically identical to a standard Euclidean activation, while the outer $\sinh/\sqrt{1+\sum\sinh^2}$ automatically projects the output back into $\mathbb{K}^m_K$. Adding ϕ contributes one elementwise evaluation per output and does not change the $\mathcal{O}(nm)$ leading-order FLOP totals above. For fairness with Chen et al. (2026a, Tab. 3)'s BFC entries, Tab. 38 reports the activation-free counts (i.e. $\phi = \mathrm{id}$).

Shared denominator.

The common denominator in Eq. (15) is the Lorentz time component introduced by the projection from the hyperboloid to the Klein ball. Since it is shared by all output coordinates, it cancels in every coordinate ratio, $y_i/y_j = \sinh(\sqrt{-K}s_i)/\sinh(\sqrt{-K}s_j)$, and therefore does not affect their relative ordering. The derivation is given in Sec. F.4.

E.3 KLEIN GYROBN

Algorithm and inference.

Alg. 1 below adapts the gyrogroup BN template of Chen et al. (2025b) to the Klein ball, using the Einstein midpoint μ_{E} of Eq. (22) as the closed-form proxy for the Fréchet mean. During training the

running mean is updated as the binary Einstein midpoint $\mu_r \leftarrow \gamma_{\mu_r}\mu_r/(\gamma_{\mu_r}+\gamma_{\mu_b}) \oplus_{\mathrm{E}} \gamma_{\mu_b}\mu_b/(\gamma_{\mu_r}+\gamma_{\mu_b})$, the Klein analogue of the running-mean barycenter of Chen et al. (2025b) Algorithm 1.

Algorithm 1 Klein Gyrogroup Batch Normalization (Klein GyroBN)

Require: batch $\{x_1,\dots,x_m\} \subset \mathbb{K}^n_{\mathrm{K}}$, running mean $\mu_r \in \mathbb{K}^n_{\mathrm{K}}$, running variance $v_r^2 \in \mathbb{R}_+$, momentum $\eta \in [0,1]$, bias $\beta \in \mathbb{K}^n_{\mathrm{K}}$, scale $\gamma \in \mathbb{R}_+$, small $\varepsilon > 0$.
Ensure: normalized batch $\{\tilde{x}_1,\dots,\tilde{x}_m\} \subset \mathbb{K}^n_{\mathrm{K}}$.
1: **if** training **then**
2: Compute batch Einstein midpoint $\mu_b = \frac{\sum_i \gamma_{x_i} x_i}{\sum_i \gamma_{x_i}}$ and batch dispersion $v_b^2 = \frac{1}{m}\sum_i d^{\mathbb{K}}((-\mu_b)\oplus_{\mathrm{E}} x_i, \mathbf{0})^2$.
3: $\mu_r \leftarrow \mathrm{Bar}^{\mathbb{K}}_{\eta}(\mu_b, \mu_r), \quad v_r^2 \leftarrow \eta\, v_b^2 + (1-\eta)\, v_r^2$.
4: **end if**
5: $(\mu, v^2) \leftarrow (\mu_b, v_b^2)$ if training else (μ_r, v_r^2).
6: **for** $i = 1,\dots,m$ **do**
7: $\tilde{x}_i \leftarrow \beta \oplus_{\mathrm{E}} \left[\frac{\gamma}{\sqrt{v^2+\varepsilon}} \otimes_{\mathrm{E}} ((-\mu)\oplus_{\mathrm{E}} x_i)\right]$.
8: **end for**

Relation between the Einstein and Fréchet means.

The Einstein midpoint μ_{E} of Eq. (22) replaces the Fréchet mean μ_{F} from Sec. 3.4 for computational efficiency in our Klein GyroBN. μ_{E} is the Klein-coordinate expression of the Lorentzian centroid of Law et al. (2019) (it solves $\arg\min_y \sum_i d^{\mathbb{L}}_{\mathcal{L}}(z_i, y)^2$ on the hyperboloid $\mathbb{L}^n_K$ under the Lorentzian distance $d^{\mathbb{L}}_{\mathcal{L}}$, not the Riemannian distance $d^{\mathbb{K}}$), so $\mu_{\mathrm{E}} \neq \mu_{\mathrm{F}}$ in general. We nevertheless identify five regimes in which the two coincide exactly or admit a quantitative approximation bound.

Proposition E.2 (Boundary equivalences and small-radius bound)**.** *Let $\{x_i\}_{i=1}^m \subset \mathbb{K}^n_{\mathrm{K}}$ with $K < 0$.*

(i) Degenerate batch. *If all $x_i = p$, then $\mu_{\mathrm{E}} = \mu_{\mathrm{F}} = p$.*

(ii) Euclidean limit. *As $K \to 0^-$, both μ_{E} and μ_{F} converge to the arithmetic mean $\bar{x} = \frac{1}{m}\sum_i x_i \in \mathbb{R}^n$.*

(iii) Binary case ($m = 2$). *For any $x_1, x_2 \in \mathbb{K}^n_{\mathrm{K}}$,*

$$\mu_{\mathrm{E}}(\{x_1,x_2\}) = \frac{\gamma_{x_1}x_1 + \gamma_{x_2}x_2}{\gamma_{x_1}+\gamma_{x_2}} = \mu_{\mathrm{F}}(\{x_1,x_2\}),$$

both equal to the unique Klein geodesic midpoint of x_1 and x_2 (Ungar, 2022, Ch. 6.18).

(iv) Origin-symmetric batch. *If the multiset $\{x_i\}_{i=1}^m$ satisfies $\{x_i\} = \{-x_i\}$ (each sample paired with its antipode), then $\sum_i \gamma_{x_i} x_i = \mathbf{0}$, hence $\mu_{\mathrm{E}} = \mathbf{0}$, and by symmetry $\mu_{\mathrm{F}} = \mathbf{0}$ as well.*

(v) Small-radius bound. *If $\|x_i\| \le \rho$ for all i with $|K|\rho^2 \ll 1$, then both means agree with $\bar{x}$ to second order,*

$$\mu_{\mathrm{E}} = \bar{x} + O(|K|\rho^2), \qquad \mu_{\mathrm{F}} = \bar{x} + O(|K|\rho^2),$$

so $d^{\mathbb{K}}(\mu_{\mathrm{E}}, \mu_{\mathrm{F}}) = O(|K|\rho^2)$.

Proof. (i) For a degenerate batch, both objective functions are minimized at $y = p$ exactly.

(ii) As $K \to 0^-$, $\gamma_{x_i} \to 1$ uniformly, so Eq. (22) reduces to $\mu_{\mathrm{E}} = \frac{1}{m}\sum_i x_i = \bar{x}$. The Klein metric reduces to the Euclidean inner product, hence $d^{\mathbb{K}}(x_i, y)^2 \to \|x_i - y\|^2$ and the Fréchet objective is minimized at $\bar{x}$.

(iii) For $m = 2$, the Fréchet objective $d^{\mathbb{K}}(x_1,y)^2 + d^{\mathbb{K}}(x_2,y)^2$ is uniquely minimized on the Klein geodesic between x_1 and x_2 at the point y_* satisfying $d^{\mathbb{K}}(x_1,y_*) = d^{\mathbb{K}}(x_2,y_*)$, the geodesic midpoint. Via the Klein–Lorentz isometry $\phi_{\mathbb{K}\to\mathbb{L}}(x) = (\gamma_x/\sqrt{-K},\, \gamma_x x)$, see Eq. (27), the Lorentz midpoint of $\phi_{\mathbb{K}\to\mathbb{L}}(x_1) + \phi_{\mathbb{K}\to\mathbb{L}}(x_2)$, normalized back to the hyperboloid and projected to Klein coordinates, gives $(\gamma_1 x_1 + \gamma_2 x_2)/(\gamma_1+\gamma_2)$. This matches the closed form of Einstein midpoint (22), confirming $\mu_{\mathrm{E}} = \mu_{\mathrm{F}}$ in this case.

(iv) The Lorentz factor depends only on the norm, $\gamma_{x_i} = \gamma_{-x_i}$, so the antipodal pairing $\{x_i\} = \{-x_i\}$ gives $\sum_i \gamma_{x_i} x_i = \frac{1}{2}\sum_i \gamma_{x_i}(x_i + (-x_i)) = \mathbf{0}$, whence $\mu_{\mathrm{E}} = \mathbf{0}$. For the Fréchet mean, the Klein metric satisfies $d^{\mathbb{K}}(-x,-y) = d^{\mathbb{K}}(x,y)$, so the map $y \mapsto -y$ is an isometry that fixes the multiset

$\{x_i\}$ and therefore the Fréchet objective. The unique minimizer must lie at the fixed point $\mathbf{0}$ of this isometry.

(v) For $\|x_i\| \le \rho$ with $|K|\rho^2 \ll 1$, Taylor expansion of the Lorentz factor gives $\gamma_{x_i} = (1 + K\|x_i\|^2)^{-1/2} = 1 + \frac{1}{2}|K|\|x_i\|^2 + O(|K|^2\rho^4)$. Substituting into Eq. (22),

$$\mu_{\mathrm{E}} = \frac{\sum_i (1 + \frac{1}{2}|K|\|x_i\|^2)\, x_i + O(|K|^2\rho^5)}{\sum_i (1 + \frac{1}{2}|K|\|x_i\|^2) + O(|K|^2\rho^4)} = \bar{x} + O(|K|\rho^2).$$

The Klein metric expands as $g^{\mathbb{K}}_x(u, v) = \langle u, v\rangle + O(|K|\rho^2)$ near the origin, so $d^{\mathbb{K}}(x, y) = \|x - y\| + O(|K|\rho^2)$ and the Fréchet objective becomes $\sum_i \|x_i - y\|^2 + O(|K|\rho^2)$, whose minimizer is $\mu_{\mathrm{F}} = \bar{x} + O(|K|\rho^2)$. The triangle inequality then gives $d^{\mathbb{K}}(\mu_{\mathrm{E}}, \mu_{\mathrm{F}}) \le \|\mu_{\mathrm{E}} - \mu_{\mathrm{F}}\|_2 + O(|K|\rho^2) = O(|K|\rho^2)$. □

In practice the empirical clip $\|x_i\| \le r/\sqrt{-K}$ used in our image-classification pipeline Sec. 4.2 gives $|K|\rho^2 = r^2 \le 9$ on Tiny-ImageNet (worst case r=3), so the small-radius regime is not attained and $\mu_{\mathrm{E}}, \mu_{\mathrm{F}}$ can differ. The empirical result that the closed-form Einstein midpoint matches the converged Fréchet mean within seed noise on the four NC datasets Sec. 4.5 shows that the approximation error is small enough not to harm downstream accuracy.

Sensitivity to boundary outliers. The Einstein midpoint assigns each point a weight γ_{x_i} that increases with radius, making the batch center more sensitive to points near the boundary. Tab. 39 measures this effect by injecting one outlier into a batch of 128 inliers and reporting the resulting geodesic drift. Within the clipped operating range, the outlier weight is bounded by $\cosh(\sqrt{-K}r)/(\cosh(\sqrt{-K}r) + m - 1)$ and is 2.9% at r=2 and m=128; at larger radii, the Einstein midpoint becomes increasingly sensitive to the outlier. For settings with unbounded outliers, the midpoint can instead initialize a subsequent Fréchet-mean iteration.

Table 39: Batch-center geodesic drift under one injected outlier, mean over 500 repetitions, K=−1, inlier radius ≤ 1, batch m=128.

r_{out}	Einstein midpoint	Fréchet mean (converged)	Tangent mean	Outlier weight share
1	0.008	0.007	0.008	1.2%
2	0.024	0.014	0.016	2.9%
4	0.156	0.028	0.031	17.7%
8	1.537	0.057	0.063	92.1%

F Proofs

F.1 Proof of Thm. 3.1

We prove the hyperplane equivalence and the point-to-hyperplane distance in turn.

Hyperplane equivalence. Set $w = \mathrm{PT}^{\mathbb{K}}_{p\to\mathbf{0}}(a) \in T_{\mathbf{0}}\mathbb{K}^n_{\mathrm{K}}$. Since parallel transport is a linear isometry,

$$\big\langle \mathrm{Log}^{\mathbb{K}}_p(x),\, a\big\rangle_p = \big\langle \mathrm{PT}^{\mathbb{K}}_{p\to\mathbf{0}}\big(\mathrm{Log}^{\mathbb{K}}_p(x)\big),\, w\big\rangle_{\mathbf{0}}. \tag{35}$$

The gyrovector identity (Ungar, 2022) gives $\mathrm{PT}^{\mathbb{K}}_{p\to\mathbf{0}}\big(\mathrm{Log}^{\mathbb{K}}_p(x)\big) = \mathrm{Log}^{\mathbb{K}}_{\mathbf{0}}\big((-p)\oplus_{\mathrm{E}} x\big)$. At the origin, $g^{\mathbb{K}}_{\mathbf{0}}(u, v) = \langle u, v\rangle$ and $\mathrm{Log}^{\mathbb{K}}_{\mathbf{0}}(z) = \frac{1}{\sqrt{-K}}\,\mathrm{arctanh}(\sqrt{-K}\|z\|)\frac{z}{\|z\|}$ is a positive scalar multiple of z. Substituting into Eq. (35),

$$\big\langle \mathrm{Log}^{\mathbb{K}}_p(x),\, a\big\rangle_p = 0 \iff \big\langle (-p)\oplus_{\mathrm{E}} x,\, w\big\rangle = 0,$$

which is Eq. (7).

Signed point-to-hyperplane distance. Set $z = (-p)\oplus_{\mathrm{E}} y$. Left Einstein translation $L_\beta(x) = \beta\oplus_{\mathrm{E}} x$ is a Klein isometry (Chen et al., 2025b, Thm. 16), so

$$d^{\mathbb{K}}(y, H_{a,p}) = d^{\mathbb{K}}\big(z,\ H_{w,\mathbf{0}}\big), \qquad H_{w,\mathbf{0}} = \{x \in \mathbb{K}^n_{\mathrm{K}} \mid \langle x, w\rangle = 0\}.$$

We compute the right-hand side via the Klein–Lorentz isometry. See Eq. (27). Define

$$z_{\mathbb{L}} = \big(\gamma_z/\sqrt{-K},\ \gamma_z\, z\big) \in \mathbb{L}^n_K, \qquad n_{\mathbb{L}} = (0,\ w) \in \mathbb{R}^{1+n}.$$

Since $x_{\mathbb{K}} = x_s/x_t$, the equation $\langle x_{\mathbb{K}}, w\rangle = 0$ on $\mathbb{K}^n_{\mathrm{K}}$ is equivalent to $\langle x_s, w\rangle = 0$ on $\mathbb{L}^n_K$, so $H_{w,\mathbf{0}}$ lifts to the totally geodesic submanifold $S = \mathbb{L}^n_K \cap \{x \mid \langle x_s, w\rangle = 0\}$ orthogonal to $n_{\mathbb{L}}$.

By the standard signed point-to-submanifold formula on the Lorentz hyperboloid,

$$d^{\mathbb{K}}(z, H_{w,\mathbf{0}}) = \frac{1}{\sqrt{-K}} \operatorname{arcsinh}\left(\frac{\sqrt{-K}\,|\langle z_{\mathbb{L}},\, n_{\mathbb{L}}\rangle_{\mathbb{L}}|}{\|n_{\mathbb{L}}\|_{\mathbb{L}}}\right).$$

A direct computation gives $\langle z_{\mathbb{L}}, n_{\mathbb{L}}\rangle_{\mathbb{L}} = \langle \gamma_z z,\, w\rangle = \gamma_z \langle z, w\rangle$ and $\|n_{\mathbb{L}}\|_{\mathbb{L}} = \|w\|$, which yields

$$d^{\mathbb{K}}(y, H_{a,p}) = \frac{1}{\sqrt{-K}} \operatorname{arcsinh}\left(\frac{\sqrt{-K}\,\gamma_z\,|\langle z, w\rangle|}{\|w\|}\right),$$

matching Eq. (8). □

F.2 Proof of Thm. 3.2

Setup. The unidirectional parameterization $p_k = \operatorname{Exp}^{\mathbb{K}}_{\mathbf{0}}(r_k \hat{z}_k)$, $a_k = \operatorname{PT}^{\mathbb{K}}_{\mathbf{0}\to p_k}(z_k)$ implies, by the isometry of parallel transport, $w_k := \operatorname{PT}^{\mathbb{K}}_{p_k\to\mathbf{0}}(a_k) = z_k$ and $\|a_k\|_{p_k} = \|z_k\|$. Substituting these into the signed-margin form Eq. (9) and using $\operatorname{sign}(u)\operatorname{arcsinh}(|u|) = \operatorname{arcsinh}(u)$,

$$v_k(x) = \frac{\|z_k\|}{\sqrt{-K}} \operatorname{arcsinh}\left(\frac{\sqrt{-K}\,\gamma_z\,\langle z, z_k\rangle}{\|z_k\|}\right), \qquad z = (-p_k) \oplus_{\mathrm{E}} x. \tag{36}$$

It remains to evaluate $\gamma_z\langle z, z_k\rangle$ in closed form.

Lift to the Lorentz model. The Klein–Lorentz isometry Eq. (27) lifts $x \in \mathbb{K}^n_{\mathrm{K}}$ to $x_{\mathbb{L}} = (\gamma_x/\sqrt{-K},\, \gamma_x x) \in \mathbb{L}^n_K$. Under this lift the foot point becomes

$$(p_k)_{\mathbb{L}} = \tfrac{1}{\sqrt{-K}}\big(\cosh(\sqrt{-K}\,r_k),\, \sinh(\sqrt{-K}\,r_k)\,\hat{z}_k\big),$$

and Einstein subtraction $z = (-p_k) \oplus_{\mathrm{E}} x$ corresponds to a Lorentz boost of $x_{\mathbb{L}}$ along $\hat{z}_k$ with rapidity $\rho = \sqrt{-K}\,r_k$.

Compute the boost. Decompose x as $x = \langle x, \hat{z}_k\rangle\,\hat{z}_k + x_\perp$. Applying the boost to $x_{\mathbb{L}}$ and reading off the spatial component along $\hat{z}_k$,

$$z_s \cdot \hat{z}_k = \gamma_x\,\langle x, \hat{z}_k\rangle \cosh(\sqrt{-K}\,r_k) - \tfrac{\gamma_x}{\sqrt{-K}} \sinh(\sqrt{-K}\,r_k).$$

The Klein–Lorentz lift also gives $z_s \cdot \hat{z}_k = \gamma_z\,\langle z, \hat{z}_k\rangle$. Multiplying both sides by $\|z_k\|$ and using $z_k = \|z_k\|\hat{z}_k$,

$$\gamma_z\,\langle z, z_k\rangle = \gamma_x\Big[\cosh(\sqrt{-K}\,r_k)\,\langle x, z_k\rangle - \tfrac{\|z_k\|}{\sqrt{-K}}\,\sinh(\sqrt{-K}\,r_k)\Big].$$

Substitute back. Dividing through by $\|z_k\|$ and inserting the result into Eq. (36),

$$\frac{\sqrt{-K}\,\gamma_z\langle z, z_k\rangle}{\|z_k\|} = \gamma_x\left(\cosh(\sqrt{-K}\,r_k)\,\frac{\sqrt{-K}}{\|z_k\|}\langle x, z_k\rangle - \sinh(\sqrt{-K}\,r_k)\right),$$

which is precisely the bracketed argument of Eq. (10).

Euclidean limit. As $K \to 0^-$, $\gamma_x \to 1$, $\cosh(\sqrt{-K}\,r_k) \to 1$, $\sinh(\sqrt{-K}\,r_k) \to \sqrt{-K}\,r_k$, and $\frac{\|z_k\|}{\sqrt{-K}}\operatorname{arcsinh}(\sqrt{-K}\,u) \to \|z_k\|\,u$. Combining, $v_k(x) \to \langle x, z_k\rangle - r_k\|z_k\|$, recovering the Euclidean MLR Eq. (6) with bias $b_k = -r_k\|z_k\|$. □

Batch-efficient computation. Eq. (10) depends on x only through the inner product $\langle x, z_k\rangle$ and the scalar $\|x\|^2$ (via γ_x), so it is implementable as a single matmul XZ over the batch. The raw form Eq. (9) would instead require materializing a $b\times C\times n$ tensor $\{(-p_k)\oplus_{\mathrm{E}} x_i\}_{i,k}$, which causes out-of-memory failures at high b or C.

F.3 Klein BMLR: derivation of the expanded form

Closed form. The Klein Busemann function $B^{v_k}_{\mathbb{K}}(x)$ is given by Eq. (12), derived in Sec. E.1 via the Klein–Lorentz isometry and cross-checked through the Klein–Poincaré isometry. Substituting into $u_k(x) = -\alpha_k B^{v_k}_{\mathbb{K}}(x) + b_k$ and expanding the log,

$$u_k(x) = \tfrac{\alpha_k}{\sqrt{-K}}\Big[\tfrac{1}{2}\log(1 + K\|x\|^2) - \log(1 - \sqrt{-K}\langle x, v_k\rangle)\Big] + b_k,$$

which is Eq. (13).

Euclidean limit. A first-order Taylor expansion as $K \to 0^-$ gives

$$\tfrac{1}{2}\log(1 + K\|x\|^2) = O(K),$$
$$\log\big(1 - \sqrt{-K}\langle x, v_k\rangle\big) = -\sqrt{-K}\langle x, v_k\rangle + O(-K).$$

Substituting, $u_k(x) \to \frac{\alpha_k}{\sqrt{-K}} \cdot \sqrt{-K}\langle v_k, x\rangle + b_k = \alpha_k\langle v_k, x\rangle + b_k$, recovering the Euclidean MLR Eq. (6) exactly. □

F.4 Proof of Thm. 3.4 (Klein FC and BFC)

Reduce defining equation at the origin. Setting $p = \mathbf{0}$ in Eq. (7) gives $H_{e_k,\mathbf{0}} = \{y \in \mathbb{K}_K^m \mid y_k = 0\}$ and the offset variable $z = (-\mathbf{0}) \oplus_{\mathrm{E}} y = y$. Applying Eq. (8) with $w = e_k$ to the defining equation (14) of $\mathcal{F}$,

$$\frac{1}{\sqrt{-K}}\operatorname{arcsinh}\big(\sqrt{-K}\,\gamma_y\, y_k\big) = s_k(x),$$

where we have absorbed the sign via $\operatorname{sign}(y_k)\operatorname{arcsinh}(|u|) = \operatorname{arcsinh}(u)$.

Solve coordinate-wise. Inverting arcsinh,

$$\gamma_y\, y_k = \frac{1}{\sqrt{-K}}\sinh\big(\sqrt{-K}\, s_k(x)\big), \qquad k = 1, \dots, m. \tag{37}$$

Solve for γ_y. Squaring and summing Eq. (37) over k,

$$\gamma_y^2\,\|y\|^2 = S, \qquad S := \tfrac{1}{-K}\sum_{j=1}^{m}\sinh^2\big(\sqrt{-K}\, s_j(x)\big).$$

Since $\gamma_y^2 = 1/(1 + K\|y\|^2)$, we have $\gamma_y^2\|y\|^2 = \|y\|^2/(1 + K\|y\|^2)$. Writing $u = \|y\|^2$, the identity $u/(1 - (-K)u) = S$ yields $u = S/(1 - KS)$ and hence

$$1 + Ku = \frac{1}{1 - KS}, \qquad \gamma_y = \sqrt{1 - KS} = \sqrt{1 + \sum_{j=1}^{m}\sinh^2\big(\sqrt{-K}\, s_j(x)\big)}.$$

Conclude. Dividing Eq. (37) by γ_y gives Eq. (15). The BFC case is identical with $s_k = u_k$ in place of v_k.

Euclidean limit. As $K \to 0^-$, $\sinh(\sqrt{-K}\, s_j) \to \sqrt{-K}\, s_j$ and $\sum_j \sinh^2(\sqrt{-K}\, s_j) \to 0$, so Eq. (15) reduces to $y_k = s_k(x)$, the Euclidean affine map. □