Umbilic Multinomial Logistic Regression

Zihan Su
Harvard University

Nicu Sebe
University of Trento

Bernhard Schölkopf
MPI for Intelligent Systems, Tübingen

Ziheng Chen[†]
University of Trento
MPI for Intelligent Systems, Tübingen

Abstract

Multinomial logistic regression (MLR) scores each class using a scaled signed distance to a class-specific hyperplane. This geometric perspective has motivated extensions of MLR to Riemannian manifolds through generalized class hyperplanes. We revisit the geometry of class hypersurfaces in Euclidean space, where the complete, connected, totally umbilic hypersurfaces are precisely hyperplanes and spheres. Standard MLR uses only hyperplanes, whereas squared-distance prototype classifiers can be viewed as the zero-radius limit of a spherical formulation. We introduce Umbilic Multinomial Logistic Regression (UMLR), which incorporates both geometries into a unified family encompassing MLR and prototype classifiers. By separating the choice of class geometry from the score link, UMLR enables controlled comparisons of hyperplanes and spheres. These comparisons reveal complementary strengths: spheres are favored in a range of low-shot settings, even with a fixed score link and without learning a radius, whereas hyperplanes are generally favored with full-data or long-tailed training. Code is available at `https://github.com/NickyoyoSu/Umbilic-Multinomial-Logistic-Regression`.

1 Introduction

Multinomial logistic regression (MLR), a fully connected layer followed by softmax [3, 27], is the standard classification layer of deep neural networks. Its logits admit a geometric reading: each affine logit is a scaled signed distance from the feature to a class hyperplane [23]. Through this point-to-hyperplane (P2H) view, MLR has been extended to many manifolds: the Poincaré ball [10, 19, 37], the Lorentz model [1, 10, 36], the proper-velocity model [12], symmetric positive definite (SPD) manifolds [7, 9, 11, 29], correlation manifolds [13], and general Riemannian manifolds [8].

Because curved spaces have no unique notion of a hyperplane, these extensions differ in two choices, summarized in Tables 1 and 2 and detailed in Appendix B. The first is the class hypersurface: most works use geodesic hyperplanes, defined through the Riemannian logarithm or cut out by linear hyperplanes of the ambient Minkowski space of the Lorentz model, whereas Busemann MLR uses horospheres, the level sets of Busemann functions [10]. The second is the P2H distance: most works use the exact P2H distance, the geodesic distance to the class hypersurface, whereas the gyrovector SPD MLR [29] and Riemannian MLR (RMLR) [8] use, in general, a pseudo P2H distance: a margin distance defined through Riemannian trigonometry. Chen et al. [10, App. C] compare the hyperbolic variants in detail.

These extensions generalize the class hyperplane to curved spaces. We revisit the Euclidean case, the original setting of MLR. A hyperplane is a totally umbilic hypersurface: its normal bending is the same in every tangent direction, namely zero. In Euclidean space of dimension at least three,

[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Table 1: Class hypersurfaces of point-to-hyperplane (P2H) MLRs. The full table, with formulas, is in Appendix B.

Hypersurface	Poincaré	Lorentz	Proper velocity	SPD	Correlation	Riemannian
Geodesic hyperplane	[19, 37]	[1, 36]	[12]	[7, 9, 11, 29]	[13]	[8]
Horosphere	[10]	[10]				

Table 2: P2H distances of these MLRs. The full table, with formulas, is in Appendix B.

P2H distance	Poincaré	Lorentz	Proper velocity	SPD	Correlation	Riemannian
Exact	[10, 19, 37]	[1, 10, 36]	[12]	[7, 9, 29]	[13]	
Pseudo				[11, 29]		[8]

the complete connected totally umbilic hypersurfaces are exactly hyperplanes and spheres [31], and MLR uses only the former. The sphere carries a center and a radius, and the squared-distance prototype classifier, widely used in few-shot recognition [28, 38, 45], is exactly its zero-radius limit. Hypersphere-based classifiers, from support vector data description to hyperspherical few-shot prototypes, likewise describe a class by a center and a radius [17, 35, 40], and hyperbolic prototype methods measure proximity with Busemann functions [20]. Used as class hypersurfaces, hyperplanes capture global directional evidence and spheres capture localized radial evidence. Both need only $O(d)$ parameters per class for d-dimensional features, whereas a general quadric surface needs $O(d^2)$.

Building on this geometry, we propose *Umbilic Multinomial Logistic Regression* (UMLR), which keeps the signed-distance form of MLR but generalizes each class hyperplane to a totally umbilic hypersurface: a hyperplane or a sphere with a learnable center and radius. UMLR contains MLR as its hyperplane member and the squared-distance prototype classifier as the zero-radius limit of its spherical member with a signed squared distance. MLR and prototype classifiers are usually developed as separate designs, related by an algebraic identity [38] rather than a shared geometry, so comparisons between them change the class geometry, the function applied to the signed distance (the score link), and the radius at the same time. UMLR places both in one family and separates these three factors.

Matched experiments show that the two geometries are complementary. Spheres are favored in a range of low-shot regimes; this advantage appears with the score link held fixed and does not require learning a nonzero radius, so it is attributable to the class geometry itself. The score link has a separate effect: in the matched low-shot study, the signed squared distance improves both geometries. With full or long-tailed training data, hyperplanes are favored, except that on ImageNet-1K [15] spheres are favored under the signed squared distance.

Contributions.

1. We introduce UMLR, a family of classifiers whose class hypersurfaces are the Euclidean totally umbilic hypersurfaces, with MLR as its hyperplane member and the squared-distance prototype classifier as the zero-radius limit of its spherical member under the signed squared distance.
2. We design matched comparisons that isolate the class geometry from the score link and the radius, and show that hyperplanes and spheres are favored in complementary regimes across shot counts and in full-data and long-tailed training.
3. We show that a spherical member of UMLR, selected on validation classes, is on par with or favored over prototype classifiers in most settings of standard few-shot benchmarks, and that the spherical member with the signed squared distance performs on par with angular-margin heads [16, 44] on ImageNet-1K.

2 Related work

Point-to-hyperplane classifiers. Multinomial logistic regression is the standard multiclass softmax model [3, 27]. Lebanon and Lafferty [23] interpret its affine logits as scaled signed distances to class

hyperplanes and use this view to adapt hyperplane margin classifiers to the multinomial manifold. Hyperbolic neural networks use P2H distances to define MLR layers in the Poincaré ball [19], which HNN++ reparameterizes [37]. Hyperbolic MLRs have also been formulated in the Lorentz model [1, 36] and the proper-velocity model [12], and MLRs have further been built on SPD manifolds through gyrovector structures [29], pullback Euclidean metrics [7, 9], and Cholesky product metrics [11], on correlation manifolds [13], and on general Riemannian manifolds [8]. Busemann neural networks derive hyperbolic MLR layers from Busemann functions, giving a point-to-horosphere interpretation [10].

Prototype and metric classifiers. Distance-based classifiers are a long-standing alternative to affine softmax heads. Nearest-class-mean classifiers compare features with class representatives [28], Prototypical Networks made class means in embedding space central to few-shot learning [38], and later work improved the metric, the solver, or the embedding [2, 32, 41]. SimpleShot showed that nearest-prototype rules on centered, L2-normalized features are difficult to beat [45]. Closest to our spherical member, HyperProto represents each few-shot class by a hypersphere with a learnable center and radius and applies softmax to a score whose zero set is that hypersphere [17]; its score subtracts a squared distance from a learned radius term, whereas UMLR applies a first-order or signed-square link to the signed distance from the sphere. UMLR places affine MLR and learned-radius spheres in one totally umbilic family, with the squared-distance prototype classifier as the zero-radius limit of its signed-square spherical member.

Angular-margin classifiers. Large-margin softmax losses enlarge the angular separation between classes [24]. SphereFace, CosFace, and ArcFace normalize features and class weights and add multiplicative angular, additive cosine, and additive angular margins, respectively [16, 25, 44], and Circle Loss reweights pair similarities [39]. These methods act on a hypersphere that all classes share as a normalization manifold, whereas UMLR gives each class its own hypersurface, a hyperplane or a Euclidean sphere with a learnable center and radius. Experiment 4 compares these two uses of spherical geometry.

Gaussian and radial-basis classifiers. The squared-distance prototype score is, up to a shared constant and scale, the log-likelihood of an isotropic, shared-variance Gaussian class-conditional model [3]. General Gaussian or radial-basis-function classifiers [5] model a bandwidth or covariance instead of a signed class radius, which makes them radial relatives of the umbilic family rather than members of it.

3 Preliminaries

A hypersurface inherits a metric and connection from its ambient space, while its second fundamental form records bending in the normal direction. It is totally umbilic when this normal bending is isotropic, with the second fundamental form equal to the induced metric multiplied by a normal vector. Hyperplanes have zero normal bending, whereas spheres bend equally in every tangent direction. Formal definitions and the surrounding geometric background are provided in Appendix A; the main construction uses the following classical Euclidean classification result.

Proposition 1 (Euclidean umbilic hypersurfaces, [31])**.** *For $d \geq 3$, the complete connected totally umbilic hypersurfaces of Euclidean space $\mathbb{R}^d$ are exactly the zero sets $\mathcal{U}_{a,b} = \{x \in \mathbb{R}^d \mid u_{a,b}(x) = 0\}$ of the following two defining functions:*

$$u_{a,b}(x) = \begin{cases} \text{hyperplane:} & \langle a, x\rangle + b & a \in \mathbb{R}^d \setminus \{0\},\ b \in \mathbb{R}, \\ \text{sphere:} & b - \|x - a\| & a \in \mathbb{R}^d,\ b > 0. \end{cases} \tag{1}$$

The hyperplane case of $\mathcal{U}_{a,b}$ is the geometry already used by ordinary MLR, and the sphere case is its curved Euclidean counterpart. This two-type classification is the design space of UMLR: the next section builds matched class coordinates from both hypersurface types so that their behavior can be compared directly.

4 UMLR classifiers

4.1 From MLR to UMLR

For ordinary Euclidean MLR, $a_k \in \mathbb{R}^d \setminus \{0\}$ and $b_k \in \mathbb{R}$ define the affine logit $u_k(x) = u_{a_k,b_k}(x) = \langle a_k, x\rangle + b_k$ [3, 27], whose zero set $\mathcal{U}_k = \mathcal{U}_{a_k,b_k}$ is the class hyperplane. The logit is the signed distance to this hyperplane, scaled by $\|a_k\|$:

$$\mu_k(x) = u_k(x) = \operatorname{sign}(u_k(x))\, \|a_k\|\, \mathrm{d}(x, \mathcal{U}_k)\,. \tag{2}$$

This is the P2H view behind the MLRs on manifolds in Tables 1 and 2.

The hyperplanes and spheres $\mathcal{U}_{a,b}$ of Eq. (1) have point-to-hypersurface distances

$$\mathrm{d}(x, \mathcal{U}_{a,b}) = \begin{cases} \text{hyperplane:} & \dfrac{|\langle a, x\rangle + b|}{\|a\|}, \\ \text{sphere:} & |b - \|x - a\||\,. \end{cases} \tag{3}$$

UMLR keeps the signed-distance construction of Eq. (2) and lets $\mathcal{U}_k$ be either totally umbilic hypersurface, with defining function $u_k = u_{a_k,b_k}$ from Eq. (1), so $u_k(x) = b_k - \|x - a_k\|$ for a sphere:

$$\mu_k^{\mathrm{UMLR}}(x) = \operatorname{sign}(u_k(x))\, \|a_k\|\, \mathrm{d}(x, \mathcal{U}_k)\,. \tag{4}$$

Substituting Eq. (3) into Eq. (4) gives the first-order UMLR forms (proof in Appendix H).

Theorem 1 (UMLR class-coordinate theorem)**.** *Let $u_{a,b}$ and $\mathcal{U}_{a,b}$ be as in Proposition 1, and let* H *and* S *denote the hyperplane and spherical cases. Then the UMLR logits of Eq.* (4) *are*

$$\begin{cases} \mu_k^{\text{UMLR-H}}(x) = \langle a_k, x\rangle + b_k, \\ \mu_k^{\text{UMLR-S}}(x) = \|a_k\|\,(b_k - \|x - a_k\|)\,. \end{cases} \tag{5}$$

The geometry determines the signed class coordinate u_k, but not its score link. Since prototype classifiers use squared Euclidean distance [28, 38], we also consider the signed-square link $z \mapsto z|z| = \operatorname{sign}(z)z^2$, which, unlike the ordinary square z^2, keeps the side of the hypersurface. Applied to the signed distance $\operatorname{sign}(u_k(x))\,\mathrm{d}(x, \mathcal{U}_k)$, with the scale $\|a_k\|$ of Eq. (4) kept outside the link, it gives the signed squared-distance logit

$$\mu_k^{\mathrm{UMLR}^2}(x) = \operatorname{sign}(u_k(x))\, \|a_k\|\, \mathrm{d}(x, \mathcal{U}_k)^2\,, \tag{6}$$

and substituting Eq. (3) gives

$$\begin{cases} \mu_k^{\text{UMLR}^2\text{-H}}(x) = \dfrac{1}{\|a_k\|} \operatorname{sign}(\langle a_k, x\rangle + b_k)\,(\langle a_k, x\rangle + b_k)^2\,, \\ \mu_k^{\text{UMLR}^2\text{-S}}(x) = \|a_k\| \operatorname{sign}(b_k - \|x - a_k\|)\,(b_k - \|x - a_k\|)^2\,. \end{cases} \tag{7}$$

UMLR^2-S is motivated by its exact zero-radius prototype limit (Section 4.2); UMLR^2-H applies the same link as a matched hyperplane counterpart. The four members together separate the score link from the class geometry.

In the experiments, the geometric scale factors, $\|a_k\|$ for the sphere and $\|a_k\|^{-1}$ for UMLR^2-H, are replaced by a class scale α_k, learned or fixed at 1 depending on the recipe (Appendix C). For UMLR-H, a positive α_k is absorbed into a_k and b_k, so UMLR-H is ordinary MLR. With $\phi(z) = z$ for the first-order link and $\phi(z) = z|z|$ for the signed-square link, every implemented logit has the form $\mu_k(x) = \alpha_k \phi(u_k(x))$; Table 3 lists all four.

Class hypersurfaces and multiclass decision boundaries. The class hypersurface $\mathcal{U}_k$ is the zero set of the k-th signed class coordinate u_k; it is not by itself a multiclass decision boundary. The pairwise equality hypersurface for classes k and j is

$$\mathcal{E}_{kj} = \{x \mid \alpha_k \phi(u_k(x)) = \alpha_j \phi(u_j(x))\}\,. \tag{8}$$

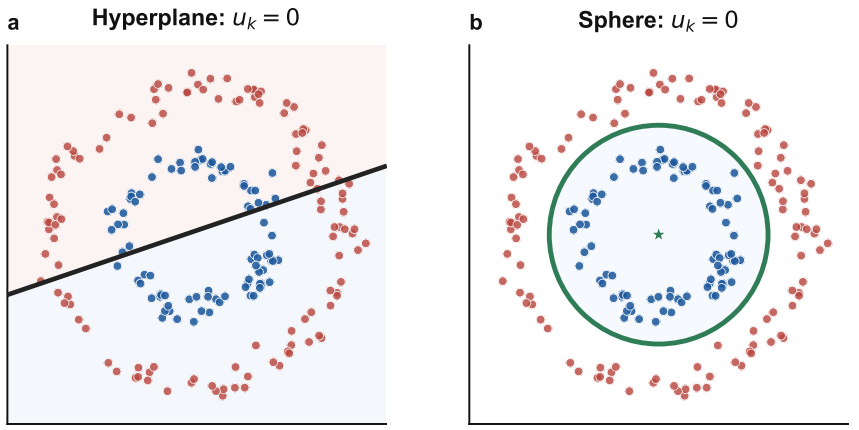

Figure 1: Hyperplane and spherical class hypersurfaces on concentric-circle toy data. Each curve is the zero set $\mathcal{U}_k$ of one class coordinate. The multiclass decision boundaries, not drawn here, are the sets $\mathcal{B}_{kj}$ of Eq. (9), the active portions of the pairwise equality hypersurfaces $\mathcal{E}_{kj}$ of Eq. (8).

Table 3: Implemented UMLR logits. H and S denote hyperplane and spherical class hypersurfaces.

Link ϕ	Surface	Class-k logit	Recovers
z	Hyperplane	$\mu_k^{\text{UMLR-H}}(x) = \langle a_k, x\rangle + b_k$	MLR
	Sphere	$\mu_k^{\text{UMLR-S}}(x) = \alpha_k(b_k - \|x - a_k\|)$	
$z\|z\|$	Hyperplane	$\mu_k^{\text{UMLR}^2\text{-H}}(x) = \alpha_k(\langle a_k, x\rangle + b_k)\|\langle a_k, x\rangle + b_k\|$	
	Sphere	$\mu_k^{\text{UMLR}^2\text{-S}}(x) = \alpha_k(b_k - \|x - a_k\|)\|b_k - \|x - a_k\|\|$	Prototype (zero-radius limit)

Because the classifier predicts $\arg\max_\ell \mu_\ell(x)$, the active decision boundary between the prediction regions of classes k and j is the subset

$$\mathcal{B}_{kj} = \{x \in \mathcal{E}_{kj} \mid \mu_k(x) = \mu_j(x) \geq \mu_\ell(x),\ \forall \ell\}\,. \tag{9}$$

For the first-order link, the pairwise equality hypersurfaces are

$$\mathcal{E}_{kj}^{\text{H}} = \{x \mid \langle \alpha_k a_k - \alpha_j a_j, x\rangle + \alpha_k b_k - \alpha_j b_j = 0\}\,, \tag{10}$$

$$\mathcal{E}_{kj}^{\text{S}} = \{x \mid \alpha_k(b_k - \|x - a_k\|) = \alpha_j(b_j - \|x - a_j\|)\}\,. \tag{11}$$

For UMLR^2, the equality in Eq. (8) reads $\alpha_k u_k(x)|u_k(x)| = \alpha_j u_j(x)|u_j(x)|$. If $\alpha_k = \alpha_j > 0$, the strict monotonicity of both links makes the equality in Eq. (8) equivalent to $u_k(x) = u_j(x)$; unequal scales change the equality hypersurface. Figure 1 shows the two class hypersurfaces on the same toy data.

4.2 Prototype as the zero-radius limit

Prototype classifiers are a successful distance-based alternative to affine softmax heads, especially in few-shot recognition [28, 38, 42, 45]. They score a feature by its negative squared distance to a class center, $\mu_k^{\text{proto}}(x) = -\|x - a_k\|^2$. For the implemented signed-square spherical logit $\alpha_k(b_k - \|x - a_k\|)|b_k - \|x - a_k\||$ (Table 3), setting $b_k = 0$ and $\alpha_k = 1$ gives

$$-\|x - a_k\|^2 = \mu_k^{\text{proto}}(x). \tag{12}$$

Thus prototype scoring is exactly the zero-radius limit of UMLR^2-S; here $b_k = 0$ is the degenerate sphere outside the $b > 0$ range of Proposition 1, and for $b_k > 0$ the same center a_k carries a sphere of radius b_k as its class hypersurface. In the geometric form of Eq. (7), where the scale is tied to $\|a_k\|$, setting $b_k = 0$ gives the class-scaled score $-\|a_k\|\,\|x - a_k\|^2$.

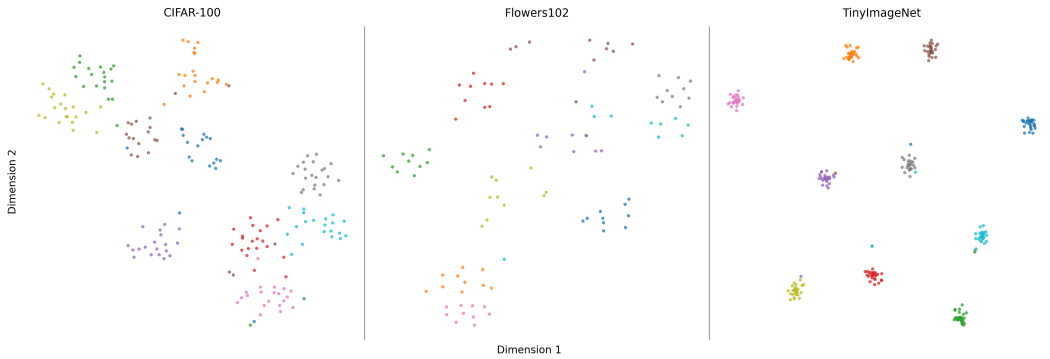

Figure 2: t-SNE of ViT-B/16 features for ten sampled classes from each dataset: frozen pretrained features for CIFAR-100 and Flowers102, and features after end-to-end fine-tuning for TinyImageNet.

5 Experiments

Datasets and backbones. Experiments 1 and 2 use the same three datasets: CIFAR-100 [21], Flowers102 [30], and TinyImageNet, a compact ImageNet-derived benchmark [15, 22]. We evaluate frozen-feature low-shot heads at 4-shot and 16-shot, and full-data heads with the backbone fine-tuned end to end (Appendix C). The two feature extractors are ViT-B/16 [18] and ConvNeXt-Tiny [26]. All main tables report accuracy in percent. Unless otherwise stated, low-shot rows use normalized frozen features so distances and radii are on a controlled scale.

Heads. We report UMLR-H, UMLR-S, UMLR^2-H, UMLR^2-S, and prototype heads. UMLR-H is ordinary MLR, and UMLR-S uses the signed distance to a class sphere. The superscript 2 denotes signed squared distance to the same class hypersurface. Prototype denotes the squared center-distance score $-\|x - a_k\|^2$; in Experiment 2 it is implemented as UMLR^2-S with every radius fixed at zero, which gives the class-scaled score $-\alpha_k \|x - a_k\|^2$. Classwise hard-gated mixtures of a hyperplane and a spherical logit (Mixed-H and Mixed-H^2) are reported in Appendix D.

5.1 Experiment 1: Hyperplane versus spherical class hypersurfaces

Comparison. UMLR-H, UMLR-S, and UMLR^2-S are trained on the same features, so the comparison isolates the class geometry and the score link. Figure 2 visualizes ViT-B/16 features for sampled classes from the three datasets: the features often form compact class-local clusters, the localized class evidence that the spherical member is designed to capture.

Table 4 shows a regime-dependent pattern. In the 12 low-shot rows, UMLR^2-S is higher than UMLR-H, most clearly on Flowers102. In the six full-data rows, UMLR-H is higher than or equal to UMLR^2-S in five, and first-order UMLR-S is lower than UMLR-H in all six.

Because Table 4 changes both the geometry and the score link, Table 5 compares all four members under a parameter-matched recipe across 1–32 shots. With the score link held fixed, the sphere is preferred at intermediate shot counts ($S^2 - H^2$ is $+2.62$ points at 2 shots and $+1.40$ at 4; $S - H$ is $+3.08$ at 4 shots and $+1.65$ at 8) and the hyperplane at 1 and 16–32 shots, so these differences are attributable to the class geometry. The signed-square link improves both geometries at every shot count. Absolute accuracies differ from Table 4 because the recipes differ.

5.2 Experiment 2: Prototype as the zero-radius limit

Comparison. Prototype is the zero-radius limit ($b_k = 0$) of UMLR^2-S. We train it as UMLR^2-S with every radius fixed at zero, so the two heads share their class-mean centers, scale, score link, and training recipe and differ only in the radius. The empirical question is whether learning a nonzero radius helps relative to this limit.

Table 6 compares UMLR^2-S with its zero-radius limit on the low-shot cells: the zero-radius limit is higher in all 12 cells, by 0.23 to 1.65 points, with paired mean differences of -0.66 points at 4

Table 4: UMLR heads with hyperplane and spherical class hypersurfaces. Accuracy entries are mean $\pm$ standard deviation over 5 seeds. All settings use the full class vocabulary. At 4/16 shots, features are frozen; full-data training fine-tunes the backbone. Differences are percentage points; Δ denotes UMLR^2-S minus UMLR-H. †Flowers102 has ten training images per class, so its 16-shot and full-data rows use all ten.

Dataset	Backbone	Regime	UMLR-H	UMLR-S	UMLR^2-S	Δ
CIFAR-100	ViT-B/16	4-shot	54.31 $\pm$ 1.38	54.69 $\pm$ 0.25	**57.44 $\pm$ 0.41**	**+3.13**
		16-shot	64.93 $\pm$ 0.35	65.56 $\pm$ 0.58	**67.50 $\pm$ 0.48**	**+2.57**
		Full data	**88.24 $\pm$0.42**	87.65 $\pm$ 0.57	88.11 $\pm$ 0.33	-0.13
	ConvNeXt-Tiny	4-shot	44.27 $\pm$ 0.60	44.88 $\pm$ 0.80	**47.98 $\pm$ 0.63**	**+3.71**
		16-shot	57.30 $\pm$ 0.65	56.08 $\pm$ 0.47	**60.76 $\pm$ 0.36**	**+3.46**
		Full data	**89.54 $\pm$ 0.39**	89.13 $\pm$ 0.44	89.42 $\pm$ 0.77	-0.12
Flowers102	ViT-B/16	4-shot	62.42 $\pm$ 0.88	66.73 $\pm$ 1.82	**70.72 $\pm$ 0.99**	**+8.30**
		16-shot†	73.37 $\pm$ 0.12	75.92 $\pm$ 0.56	**80.65 $\pm$ 0.09**	**+7.29**
		Full data†	**94.31 $\pm$ 0.21**	87.16 $\pm$ 0.85	**94.31 $\pm$ 0.69**	+0.00
	ConvNeXt-Tiny	4-shot	70.88 $\pm$ 0.76	76.11 $\pm$ 1.45	**78.30 $\pm$ 1.17**	**+7.42**
		16-shot†	81.80 $\pm$ 0.17	84.15 $\pm$ 0.33	**87.35 $\pm$ 0.08**	**+5.56**
		Full data†	95.49 $\pm$ 0.39	87.25 $\pm$ 0.24	**96.18 $\pm$ 0.58**	**+0.69**
TinyImageNet	ViT-B/16	4-shot	67.09 $\pm$ 0.74	69.22 $\pm$ 0.58	**70.25 $\pm$ 0.76**	**+3.15**
		16-shot	77.44 $\pm$ 0.49	76.37 $\pm$ 0.19	**78.61 $\pm$ 0.26**	**+1.17**
		Full data	**86.02 $\pm$ 0.64**	81.99 $\pm$ 0.51	82.31 $\pm$ 0.24	-3.71
	ConvNeXt-Tiny	4-shot	53.02 $\pm$ 0.39	53.41 $\pm$ 0.41	**56.42 $\pm$ 0.22**	**+3.40**
		16-shot	67.17 $\pm$ 0.08	64.57 $\pm$ 0.18	**70.08 $\pm$ 0.05**	**+2.91**
		Full data	**87.08 $\pm$ 0.34**	85.36 $\pm$ 0.53	85.02 $\pm$ 0.05	-2.06

Table 5: Parameter-matched shot-regime attribution. Head columns are mean accuracies (%) over three datasets, two backbones, and five paired seeds. Effect columns are percentage-point paired differences with 95% hierarchical-bootstrap confidence intervals. Flowers102 has ten training images per class, so its 16- and 32-shot cells use all ten.

Shots	UMLR-H	UMLR^2-H	UMLR-S	UMLR^2-S	$S - H$ [95% CI]	$S^2 - H^2$ [95% CI]
1	37.32	38.09	28.86	36.21	-8.47 [-11.08, -5.90]	-1.88 [-6.23, $+0.32$]
2	47.88	48.76	48.14	51.38	$+0.26$ [-1.75, $+2.40$]	$+\mathbf{2.62}$ [$+0.93$, $+4.44$]
4	58.80	62.02	61.88	63.43	$+\mathbf{3.08}$ [$+1.48$, $+4.89$]	$+1.40$ [$+0.41$, $+2.42$]
8	66.56	71.00	68.21	70.54	$+1.65$ [$+0.47$, $+2.93$]	-0.46 [-0.87, -0.07]
16	72.45	75.14	70.75	73.95	-1.69 [-3.62, $+0.02$]	-1.19 [-2.05, -0.39]
32	75.71	77.60	71.57	76.12	-4.15 [-7.36, -0.88]	-1.48 [-2.43, -0.51]

shots (95% t-CI $[-0.87, -0.46]$) and -0.82 at 16 shots ($[-1.02, -0.62]$). The parameter-matched recipe gives the same picture: with class-mean centers and the score link, scale, split, and optimizer held fixed, a learned q80 radius changes accuracy relative to a fixed $r = 0$ by -0.78 points at 4 shots (95% t-CI $[-1.07, -0.49]$) and -0.80 at 16 shots ($[-1.17, -0.43]$), with $r = 0$ higher in all 30 paired runs (ViT-B/16, three datasets, five seeds). The low-shot spherical gains therefore come from the class geometry and the score link, and the zero-radius limit already attains them; learning a nonzero radius does not help in these settings. One reading is that the class score still decreases monotonically with the distance to the center under either link, so the radius mainly shifts each class score, much like a classwise bias. Consistent with this reading, the learned radii of UMLR-S correlate with class frequency under class imbalance (Spearman 0.94 and 0.63; Appendix F), whereas with balanced classes and few examples per class such an offset has little to contribute.

5.3 Experiment 3: Validation-selected episodic adaptation

To test the family beyond the three datasets of Experiments 1 and 2, we use CIFAR-FS [2], CUB [43], FC100 [32], miniImageNet [42], and tieredImageNet [34]. To align this evaluation with the all-class low-shot design of Experiments 1 and 2, every episode uses the full class vocabulary of its split and 1/4/16 support examples per class, with 15 queries per class. All methods use the same ImageNet-pretrained ViT-B/16 features and episode draws, with no method-specific meta-training, and the UMLR heads are adapted on each support set under the same optimization budget (Appendix E).

Comparison. Reporting all four UMLR members beside the baselines would give the family four opportunities on the test set. We therefore select a single member on validation classes only. For each

Table 6: Prototype versus UMLR^2-S on the low-shot cells of Table 4. Accuracy entries are mean $\pm$ standard deviation over 5 seeds. Differences are percentage points; Δ denotes UMLR^2-S minus Prototype. †Flowers102 has ten training images per class, so its 16-shot rows use all ten.

Dataset	Backbone	Regime	Prototype	UMLR^2-S	Δ
CIFAR-100	ViT-B/16	4-shot	**57.47 ± 1.15**	56.86 ± 1.00	-0.62
		16-shot	**67.91 ± 0.49**	67.40 ± 0.44	-0.50
	ConvNeXt-Tiny	4-shot	**47.52 ± 1.10**	47.20 ± 1.24	-0.32
		16-shot	**61.10 ± 0.49**	60.52 ± 0.58	-0.58
Flowers102	ViT-B/16	4-shot	**72.08 ± 0.45**	70.65 ± 0.87	-1.43
		16-shot†	**82.53 ± 0.18**	80.88 ± 0.37	-1.65
	ConvNeXt-Tiny	4-shot	**78.78 ± 0.94**	77.82 ± 1.20	-0.96
		16-shot†	**88.98 ± 0.34**	87.57 ± 0.31	-1.41
TinyImageNet	ViT-B/16	4-shot	**70.72 ± 0.77**	70.29 ± 0.85	-0.43
		16-shot	**78.79 ± 0.26**	78.54 ± 0.25	-0.25
	ConvNeXt-Tiny	4-shot	**56.64 ± 0.35**	56.41 ± 0.28	-0.23
		16-shot	**70.65 ± 0.05**	70.13 ± 0.07	-0.53

Table 7: Validation-selected episodic comparison. Entries are mean $\pm$ sample standard deviation over five new test seeds (%). Only UMLR-S, selected globally using validation data, is compared with the baselines; bold marks the highest unrounded mean.

Dataset	Shot	ProtoNet	SimpleShot-CL2N	Selected UMLR-S
CIFAR-FS	1	63.21±0.52	**68.57±0.56**	68.56±0.59
	4	81.98±0.21	81.75±0.27	**82.24±0.14**
	16	87.09±0.15	86.76±0.20	**88.33±0.20**
CUB	1	42.58±0.14	48.14±0.18	**48.15±0.19**
	4	65.53±0.07	67.34±0.04	**71.77±0.09**
	16	75.58±0.12	75.97±0.11	**83.93±0.05**
FC100	1	39.18±0.08	41.55±0.24	**41.58±0.26**
	4	56.77±0.17	56.78±0.26	**57.46±0.15**
	16	66.55±0.32	66.35±0.30	**68.61±0.15**
miniImageNet	1	73.25±0.49	**91.90±0.22**	91.90±0.22
	4	**96.72±0.09**	96.06±0.19	96.25±0.17
	16	**97.78±0.01**	97.09±0.05	97.50±0.02
tieredImageNet	1	73.28±0.09	**82.86±0.15**	82.86±0.14
	4	**90.95±0.02**	90.76±0.00	90.45±0.02
	16	**92.77±0.04**	92.37±0.03	92.46±0.05

of the 15 dataset$\times$shot settings, we run five seeds and 200 episodes per seed on the validation split; the equal-cell validation mean uniquely selects UMLR-S among H/H^2/S/S^2. We lock that member before evaluating it on disjoint test classes with new seeds, against ProtoNet and SimpleShot-CL2N on identical episode draws.

Table 7 compares the selected UMLR-S with the two prototype baselines. At 1 shot, UMLR-S and SimpleShot are nearly tied (within 0.03 points) and both exceed ProtoNet. At 4 and 16 shots, UMLR-S is preferred on CIFAR-FS, CUB, and FC100, most clearly on 16-shot CUB (83.93% versus 75.97% for SimpleShot), whereas ProtoNet is preferred on miniImageNet and tieredImageNet, where all three methods exceed 90%. Overall, UMLR-S matches or outperforms both baselines in 11 of the 15 settings; Appendix E compares all four members on the test classes.

5.4 Experiment 4: ImageNet-1K and angular-margin controls

Comparison. To test the family at a larger class count and against angular-margin heads, we discard the original ImageNet classifier and train UMLR-H, UMLR^2-H, UMLR-S, UMLR^2-S, CosFace, and ArcFace heads from scratch on cached penultimate features of all 1,281,167 ImageNet-1K training images [15]. All heads use the same frozen, supervised ImageNet-1K-pretrained ViT-B/16 representation, centered by the training mean and L2-normalized, and are trained for 20 epochs with AdamW, a cosine schedule, learning rate 10^{-3}, weight decay 10^{-4}, label smoothing 0.1, batch size 4096, and five seeds. CosFace and ArcFace use scale 30 and margin 0.35. Because the representation was pretrained on ImageNet-1K, this experiment retrains heads on the same task. Every head is

Table 8: ImageNet-1K same-task cached-feature head retraining with a matched frozen ViT-B/16 representation. Top-1 is mean $\pm$ sample standard deviation over five seeds; Top-5 is the five-seed mean. Accuracies are on the 50,000 validation images after the final training epoch.

Head	Top-1 (%)	Top-5 (%)
UMLR-H	79.95 ± 0.01	94.93
UMLR^2-H	79.64 ± 0.03	94.84
UMLR-S	79.03 ± 0.02	94.29
UMLR^2-S	80.03 ± 0.01	94.91
CosFace	79.96 ± 0.00	94.09
ArcFace	80.06 ± 0.01	94.31

evaluated on the 50,000 validation images after its final epoch; these images are not used for epoch or checkpoint selection.

Table 8 shows that all six heads lie within about one point at $K = 1{,}000$. The geometry contrast again depends on the score link: $S - H = -0.92$ points (95% t-CI $[-0.97, -0.87]$) and $S^2 - H^2 = +0.39$ points ($[+0.31, +0.46]$). The best spherical head, UMLR^2-S (80.03%), is on par with ArcFace (80.06%), CosFace (79.96%), and UMLR-H (79.95%), whereas first-order UMLR-S is about one point lower. Appendix F reports large-K computation, frozen-versus-fine-tuned, and full-data/long-tail controls.

6 Conclusion

This paper introduced UMLR, which generalizes the class hyperplanes of MLR to the totally umbilic hypersurfaces of Euclidean space, hyperplanes and spheres. The family contains MLR as its hyperplane member and the squared-distance prototype classifier as the zero-radius limit of its signed-square spherical member, so affine and distance-based classification become members of one geometric family. As class hypersurfaces, hyperplanes capture global directional evidence and spheres capture localized radial evidence, each with $O(d)$ parameters per class.

Because UMLR separates the class geometry, the score link, and the radius, these factors can be compared under matched parameterizations. The two geometries are complementary: S is favored in a range of low-shot regimes, an advantage that holds with the score link fixed and without learning a nonzero radius, whereas H is favored with full or long-tailed training data, except on ImageNet-1K, where the signed-square link favors S. In the matched low-shot study, the signed-square link improves both geometries at every shot count. Against external baselines, the validation-selected UMLR-S matches or outperforms ProtoNet and SimpleShot in most settings of five standard few-shot benchmarks, and UMLR^2-S performs on par with CosFace and ArcFace on ImageNet-1K.

Most of our experiments train classifier heads on pretrained features; extending the comparison to end-to-end representation learning and to other modalities is a natural next step for the umbilic view of classification.

7 Limitations and broader impact

Limitations. The study is head-level: most comparisons fit classifier heads on shared pretrained features, and the fine-tuning control shows that end-to-end training interacts with the score link, so the preferred geometry depends on the score link and the training protocol. In the balanced full-data controls every geometry contrast favors H, and on CIFAR-100-LT the first-order spherical head deteriorates sharply on rare classes (Appendix F). In the tested low-shot settings, learning a nonzero radius does not improve over the zero-radius limit (Experiment 2), so settings in which the radius itself is beneficial remain to be identified. The spherical head UMLR^2-S is also more sensitive to the initial scale (up to 3.30 points) than to the radius initialization (at most 0.30) or the radius learning rate (at most 0.92; Appendix C). Classwise mixed heads did not reveal an interpretable division between H and S and are reported only as an exploratory variant (Appendix D). Finally, the evidence covers image classification with ViT-B/16 and ConvNeXt-Tiny backbones; other modalities, larger backbones, and meta-trained embeddings remain untested.

Broader impact. This work analyzes the geometry of classifier heads on public image benchmarks. It does not introduce a new dataset, a deployed system, or a generative model, and we do not foresee direct negative societal impacts beyond those of image classification in general.

Acknowledgments and Disclosure of Funding

This work was supported by the EU Horizon Europe projects ELIAS (No. 101120237) and ELLIOT (No. 101214398), and by the FIS project GUIDANCE (No. FIS2023-03251). We acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LEONARDO, hosted by CINECA (Italy) and the LEONARDO consortium through the EuroHPC Development Access (EUHPC_D34_153) and AI Factory (AIFAC_S07_008) access calls. The authors declare no competing interests.

References

[1] Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=ekz1hN5QNh`.

[2] Luca Bertinetto, João F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HyxnZh0ct7`.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006. ISBN 9780387310732.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461, 2014.

[5] David S. Broomhead and David Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[6] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

[7] Ziheng Chen, Yue Song, Gaowen Liu, Ramana Rao Kompella, Xiao-Jun Wu, and Nicu Sebe. Riemannian multinomial logistics regression for SPD neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17086–17096, 2024.

[8] Ziheng Chen, Yue Song, Rui Wang, Xiao-Jun Wu, and Nicu Sebe. RMLR: Extending multinomial logistic regression into general geometries. In *Advances in Neural Information Processing Systems*, volume 37, pages 59716–59754, 2024.

[9] Ziheng Chen, Yue Song, Tianyang Xu, Zhiwu Huang, Xiao-Jun Wu, and Nicu Sebe. Adaptive log-Euclidean metrics for SPD matrix learning. *IEEE Transactions on Image Processing*, 33: 5194–5205, 2024. doi: 10.1109/TIP.2024.3451930.

[10] Ziheng Chen, Bernhard Schölkopf, and Nicu Sebe. Hyperbolic Busemann neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[11] Ziheng Chen, Yue Song, Xiao-Jun Wu, and Nicu Sebe. Fast and stable Riemannian metrics on SPD manifolds via Cholesky product geometry. In *International Conference on Learning Representations*, 2026.

[12] Ziheng Chen, Zihan Su, Bernhard Schölkopf, and Nicu Sebe. Proper velocity neural networks. In *International Conference on Learning Representations*, 2026.

[13] Ziheng Chen, Xiao-Jun Wu, Bernhard Schölkopf, and Nicu Sebe. Riemannian networks over full-rank correlation matrices. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 17467–17507, 2026.

[14] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. URL https://ieeexplore.ieee.org/abstract/document/5206848/.

[16] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[17] Ning Ding, Yulin Chen, Ganqu Cui, Xiaobin Wang, Haitao Zheng, Zhiyuan Liu, and Pengjun Xie. Few-shot classification with hypersphere modeling of prototypes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 895–917. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.57.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[19] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/dbab2adc8f9d078009ee3fa810bea142-Paper.pdf.

[20] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. Hyperbolic Busemann learning with ideal prototypes. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[22] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Technical report, Stanford University, CS 231N, 2015.

[23] Guy Lebanon and John Lafferty. Hyperplane margin classifiers on the multinomial manifold. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 66, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015333. URL https://doi.org/10.1145/1015330.1015333.

[24] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *International Conference on Machine Learning*, pages 507–516, 2016.

[25] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 212–220, 2017.

[26] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11966–11976, 2022. doi: 10.1109/CVPR52688.2022.01167.

[27] Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In Paul Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic press, New York, 1974.

[28] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriella Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2624–2637, November 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.83. URL https://doi.org/10.1109/TPAMI.2013.83.

[29] Xuan Son Nguyen and Shuo Yang. Building neural networks on matrix manifolds: A gyrovector space approach. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 26031–26062. PMLR, 2023.

[30] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[31] Barrett O'Neill. *Semi-Riemannian Geometry with Applications to Relativity*. Academic Press, 1983.

[32] Boris N. Oreshkin, Pau Rodriguez, and Alexandre Lacoste. TADAM: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, volume 31, pages 719–729. Curran Associates, Inc., 2018.

[33] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[34] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *Proceedings of 6th International Conference on Learning Representations ICLR*, 2018.

[35] Lukas Ruff, Robert Vandermeulen, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4393–4402, 2018.

[36] Xianglong Shi, Ziheng Chen, Yunhan Jiang, and Nicu Sebe. Intrinsic Lorentz neural network. In *International Conference on Learning Representations*, 2026.

[37] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Ec85b0tUwbA`.

[38] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 4077–4087, 2017. URL `https://arxiv.org/abs/1703.05175`.

[39] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020.

[40] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Machine Learning*, 54 (1):45–66, 2004.

[41] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV*, pages 266–282, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58567-9. doi: 10.1007/978-3-030-58568-6_16. URL `https://doi.org/10.1007/978-3-030-58568-6_16`.

[42] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf`.

[43] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[44] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[45] Yan Wang, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. SimpleShot: Revisiting nearest-neighbor classification for few-shot learning, 2019. URL `https://arxiv.org/abs/1911.04623`.

A Additional geometry background

The main text retains only the classical Euclidean classification result used by UMLR. We recall the surrounding definitions here for completeness, following O'Neill's notation [31, Chs. 1,3,4].

Vector fields and functions. Let $C^\infty(M)$ be the ring of smooth real-valued functions on M, and let $\mathfrak{X}(M)$ be the set of smooth vector fields. A vector field $V \in \mathfrak{X}(M)$ assigns a tangent vector $V_p \in T_pM$ to each point p. Since tangent vectors act as derivations on smooth functions, V acts on $f \in C^\infty(M)$ by

$$(Vf)(p) = V_p(f). \tag{13}$$

Thus Vf is a smooth scalar function, not a vector field. This is why terms such as Vf appear in the product rule for covariant derivatives. Smooth functions also multiply vector fields pointwise:

$$(fV)_p = f(p)V_p. \tag{14}$$

Consequently $\mathfrak{X}(M)$ is a module over $C^\infty(M)$, rather than only a vector space over $\mathbb{R}$.

Connections. A connection is the operation that differentiates one vector field in the direction of another vector field. Formally, it is a map

$$D : \mathfrak{X}(M) \times \mathfrak{X}(M) \to \mathfrak{X}(M), \qquad (V, W) \mapsto D_V W. \tag{15}$$

Thus the input is a pair of vector fields and the output is again a vector field. O'Neill's connection axioms can be written as

$$D_{fV+gW}X = fD_VX + gD_WX, \tag{16}$$
$$D_V(aX + bY) = aD_VX + bD_VY, \tag{17}$$
$$D_V(fX) = (Vf)X + fD_VX, \tag{18}$$

for $f, g \in C^\infty(M)$, $a, b \in \mathbb{R}$, and $V, W, X, Y \in \mathfrak{X}(M)$. The first line says that the differentiating direction is $C^\infty(M)$-linear. The second line says that the field being differentiated is $\mathbb{R}$-linear. The third line is the Leibniz rule: the scalar function f varies along V, producing the scalar derivative Vf, while X is still a vector field.

Levi-Civita connection. On a Riemannian manifold (M, g), the Levi-Civita connection is the unique connection D that is torsion-free and compatible with the metric:

$$D_XY - D_YX = [X, Y], \qquad X(g(Y, Z)) = g(D_XY, Z) + g(Y, D_XZ). \tag{19}$$

This is the connection used to define covariant derivatives, parallel vector fields, geodesics, parallel transport, and the exponential map [31, Ch. 3].

Induced connection and second fundamental form. If $N \subset \bar{M}$ is a Riemannian submanifold, its metric is the restriction of the ambient metric to tangent spaces. At each point, the ambient tangent space decomposes into tangent and normal parts,

$$T_p\bar{M} = T_pN \oplus T_pN^\perp. \tag{20}$$

Let $\bar{D}$ be the Levi-Civita connection of the ambient manifold. If $V, W \in \mathfrak{X}(N)$ are tangent vector fields, then the ambient derivative $\bar{D}_V W$ need not remain tangent to N. O'Neill's induced connection construction separates it into

$$\bar{D}_V W = \tan \bar{D}_V W + \operatorname{nor} \bar{D}_V W = D_V W + \mathrm{II}(V, W). \tag{21}$$

The tangent projection $D_V W = \tan \bar{D}_V W$ is the Levi-Civita connection induced on N, while the normal projection $\mathrm{II}(V, W) = \operatorname{nor} \bar{D}_V W$ is the second fundamental form. This decomposition relates intrinsic motion along the surface to extrinsic bending in the ambient feature space [31, Ch. 4].

Hypersurfaces and total umbilicity. A hypersurface is a codimension-one submanifold: if the ambient manifold has dimension d, then the hypersurface has dimension $d - 1$. A point is umbilic when its normal bending has no preferred tangent direction:

Definition 1 (Totally umbilic hypersurface, [31])**.** *Let $N \subset \bar{M}$ be a hypersurface. A point $p \in N$ is umbilic if there exists a normal vector $z \in T_pN^\perp$ such that $\mathrm{II}(v, w) = g_p(v, w)z$ for all $v, w \in T_pN$. The hypersurface is totally umbilic if every point is umbilic.*

Thus the second fundamental form is the metric multiplied by a normal vector. Hyperplanes have no normal bending, whereas spheres bend identically in every tangent direction. Proposition 1 states the corresponding complete Euclidean classification used by UMLR.

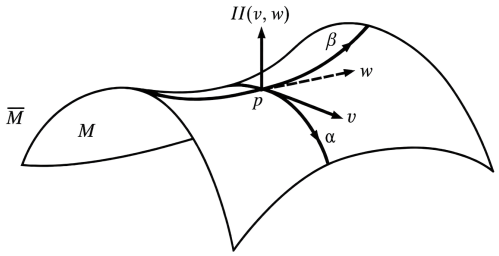

Figure 3: Normal bending at a point $p \in N$. The point is umbilic when the normal components $\mathrm{II}(v, w)$ are determined by one normal vector z, i.e., $\mathrm{II}(v, w) = g_p(v, w)z$.

B Point-to-hyperplane MLRs on manifolds

Tables 9 and 10 give the formulas behind Tables 1 and 2. Each MLR keeps the template of Eq. (2): its logit is a signed distance from the feature to a class hypersurface, scaled by a class-specific factor. We follow the notation of the cited works: p and P are base points; a, A, w, and Z are normal vectors; $\oplus$ and $\ominus$ are the gyro-operations of the respective model; B is a Busemann function; and ϕ is a diffeomorphism that pulls back a Euclidean metric, with differential ϕ_*. A P2H distance is exact when it equals the geodesic point-to-set distance $\inf_{y \in H} \mathrm{d}(x, y)$, and pseudo otherwise.

For correlation matrices, the four flat metrics (ECM, LECM, OLM, LSM) are pullback Euclidean metrics, and Z is expressed in the coordinates of ϕ; under the curved PHCM metric, the Poincaré MLR of the first row is applied after mapping to a Poincaré ball [13]. RMLR replaces the geodesic infimum by a Riemannian margin distance built from Riemannian trigonometry, where Y^* maximizes $\cos \angle SPY$ over $Y \in H$; its closed form is the tangent-space distance from $\mathrm{Log}_P S$ to $A^\perp$, which coincides with the exact P2H distance under pullback Euclidean metrics [8].

On SPD manifolds, Nguyen and Yang [29] define the class hypersurface through the Riemannian logarithm and the distance through Riemannian trigonometry, a pseudo-gyrodistance. It equals the exact distance under the Log-Euclidean and Log-Cholesky metrics, which are pullback Euclidean metrics (their Thms. 2.23–2.24), and is pseudo under the affine-invariant metric. Chen et al. [11] use

Table 9: Class hypersurfaces of the P2H MLRs in Table 1.

Manifold	Class hypersurface H
Geodesic hyperplane	
Poincaré [19, 37]	$\{x \mid \langle (-p) \oplus_c x, a \rangle = 0\} = \exp_p(\{a\}^\perp)$
Lorentz [1, 36]	$\left\{x \mid \langle w, x \rangle_{\mathcal{L}} = 0\right\}, w \in T_p \mathbb{L}^n_K$
Proper velocity [12]	$\left\{x \mid \left\langle \mathrm{Log}_p x, a \right\rangle_p = 0\right\}$
SPD [7, 9, 29] pullback Euclidean metric	$\{S \mid \langle \phi(S) - \phi(P), \phi_{*,P}(A) \rangle = 0\}$
SPD [11, 29] affine-invariant or Cholesky product metric	$\left\{S \mid \langle \mathrm{Log}_P S, A \rangle_P = 0\right\}$
Correlation [13] pullback Euclidean metric	$\{C \mid \langle \phi(C), Z \rangle - \gamma \, \|Z\| = 0\}$
Riemannian [8]	$\left\{S \mid \langle \mathrm{Log}_P S, A \rangle_P = 0\right\}$
Horosphere	
Poincaré, Lorentz [10]	$\{x \mid B^v(x) = \tau\}$

Table 10: P2H distances of the MLRs in Table 2.

Manifold	P2H distance $\mathrm{d}(x, H)$
Exact P2H distance	
Poincaré [19, 37]	$\frac{1}{\sqrt{c}} \sinh^{-1}\left(\frac{2\sqrt{c}\,\lvert\langle (-p) \oplus_c x, a\rangle\rvert}{\left(1 - c\,\lVert (-p) \oplus_c x\rVert^2\right) \lVert a\rVert} \right)$
Lorentz [1, 36]	$\frac{1}{\sqrt{-K}} \left\lvert \sinh^{-1}\left(\sqrt{-K}\, \frac{\langle w, x\rangle_{\mathcal{L}}}{\lVert w\rVert_{\mathcal{L}}} \right) \right\rvert$
Proper velocity [12]	$\frac{1}{\sqrt{c}} \sinh^{-1}\left(\frac{\sqrt{c}\,\lvert\langle (-p) \oplus_{\mathcal{PV}} x, a\rangle\rvert}{\lVert a\rVert} \right)$, $a \in T_0$
SPD [7, 9, 29] pullback Euclidean metric	$\frac{\lvert\langle \phi(S) - \phi(P), \phi_{*,P}(A)\rangle\rvert}{\lVert A\rVert_P}$
Correlation [13] pullback Euclidean metric	$\frac{\lvert\langle \phi(C), Z\rangle - \gamma\,\lVert Z\rVert\rvert}{\lVert Z\rVert}$
Poincaré, Lorentz [10] horosphere	$\lvert B^v(x) - \tau\rvert$
Pseudo P2H distance	
Riemannian [8]	$\sin(\angle SPY^*)\, \mathrm{d}(S, P) = \frac{\lvert\langle \mathrm{Log}_P S, A\rangle_P\rvert}{\lVert A\rVert_P}$
SPD [11, 29] affine-invariant or Cholesky product metric	$\frac{\lvert\langle \mathrm{Log}_P S, A\rangle_P\rvert}{\lVert A\rVert_P}$

the RMLR margin distance under power-Cholesky and Bures–Wasserstein–Cholesky metrics; these are flat but map onto a proper subset of a Euclidean space, so the margin distance is pseudo there.

C Main experiment details

Datasets and splits. The main experiments use CIFAR-100, Flowers102, and TinyImageNet. CIFAR-100 uses the standard train/test split. Flowers102 uses the training split for training and the validation split for evaluation. Because its training split contains ten images per class, the 16-shot and full-data rows of Flowers102 (marked by †) use all ten images per class, as do its 16- and 32-shot cells in Table 5. TinyImageNet uses the official training folders and the validation images reorganized by class. For low-shot rows, the support set is sampled classwise from the training split with the run seed: 4-shot uses four examples per class and 16-shot uses sixteen examples per class. Full-data rows use all available training images.

Backbones and preprocessing. The main tables use ImageNet-pretrained ViT-B/16 and ConvNeXt-Tiny from torchvision. The classifier layer of the pretrained model is removed and replaced by the head under study. Images are resized or cropped to 224×224 and normalized by ImageNet mean and standard deviation. Training uses random resized crop, horizontal flip, and RandAugment with two operations and magnitude 9; evaluation uses deterministic resize to 224×224, which is also the only transform used to cache the ImageNet-1K and episodic features. In low-shot rows the backbone is frozen. In every suite, including the fine-tuned full-data rows, features are centered by the training-set mean and then L2-normalized before the head; for fine-tuned rows this mean is computed once with the pretrained backbone before training.

Optimization. All heads are trained with cross-entropy and AdamW. Except for the episodic heads, training uses label smoothing 0.1 and a cosine schedule, with a linear warmup in every suite except the ImageNet-1K runs. The episodic heads use no label smoothing and a constant learning rate (Appendix E). Table 11 lists the settings used by the result families that feed the main tables. The head learning-rate factor multiplies the base learning rate only for parameters in the classifier head. Every epoch is evaluated on the evaluation split, and the head-swap, mixed-head, full-data, and parameter-matched tables report the best-epoch accuracy. The ImageNet-1K heads (Experiment 4)

are instead reported after the final epoch, and the episodic heads are adapted for a fixed number of steps without checkpoint selection.

Table 11: Training settings for the main experiment tables.

Result family	Regime	Epochs	Backbone	LR	Weight decay	Warmup
Head-swap suite	4-shot, 16-shot	60	frozen	10^{-3}	10^{-4}	3
Mixed-head suite	4-shot, 16-shot	60	frozen	10^{-3}	0.05	5
Full-data suite	full data	100	trainable	5×10^{-5}	0.05	5

The low-shot rows of Table 4 come from the head-swap suite, and its full-data rows from the full-data suite. The prototype comparison in Table 6 is a separate five-seed run with the head-swap recipe that pairs UMLR^2-S with its zero-radius limit seed by seed, so its UMLR^2-S column differs slightly from the same cells of Table 4. The low-shot rows of Table 13 use the mixed-head suite, whose weight decay and warmup differ from the head-swap suite, so its low-shot single-geometry columns also differ slightly from Table 4; its full-data rows come from the full-data suite and repeat the UMLR-H and UMLR^2-S entries of Table 4. The parameter-matched attribution study (Table 5) uses the parameter-matched recipe described below.

The batch size is 64 in the head-swap, mixed-head, and full-data suites; the ImageNet-1K heads of Experiment 4 use 4096. Low-shot rows use head learning-rate factor 1.0. Full-data rows use head learning-rate factor 10.0 so that the newly initialized head can move faster than the pretrained feature extractor.

Head parameterizations. The head names follow Section 4. The implemented hyperplane score is $\langle a_k, x\rangle + b_k$. UMLR-H uses this affine score, and UMLR^2-H uses $(\langle a_k, x\rangle + b_k)|\langle a_k, x\rangle + b_k|$. In the head-swap, full-data, and mixed-head suites the hyperplane heads have no separate scale; in the parameter-matched recipe and the episodic and ImageNet-1K runs, a learned classwise positive scale α_k multiplies the logit of all four heads, so that UMLR-H and UMLR^2-H become $\alpha_k(\langle a_k, x\rangle + b_k)$ and $\alpha_k(\langle a_k, x\rangle + b_k)|\langle a_k, x\rangle + b_k|$. The spherical heads use center a_k, radius $b_k > 0$, and scale α_k. The radius is represented by an unconstrained raw value mapped to a positive value by softplus in the code. With the inside-positive convention, UMLR-S uses $\alpha_k(b_k - \|x - a_k\|)$, and UMLR^2-S uses $\alpha_k(b_k - \|x - a_k\|)|b_k - \|x - a_k\||$. In the head-swap, full-data, and mixed-head suites, UMLR^2-S learns α_k directly, without a positivity constraint and from an initial value of 1, whereas UMLR-S keeps $\alpha_k > 0$ through softplus and starts from $\alpha_k = \ln 2 \approx 0.69$, the softplus of a zero raw value.

The prototype head in Table 6 is UMLR^2-S with every radius fixed at zero, $-\alpha_k \|x - a_k\|^2$; it shares the class-mean centers, scale, and training recipe of UMLR^2-S.

Initialization. Hyperplane weights are drawn from $\mathcal{N}(0, 1/d)$ and biases are initialized at zero. In Mixed-H and Mixed-H^2 the hyperplane weights are instead drawn from $\mathcal{N}(0, 0.02^2)$, and the spherical-side scale α_k starts at the norm of that class's initial hyperplane weight. Spherical heads initialize each center at the mean training feature of its class and each radius at a quantile of the within-class distances to that center: the 0.9 quantile in the head-swap, full-data, and mixed-head suites and the 0.8 quantile in the parameter-matched recipe and in the ImageNet-1K heads of Experiment 4. The prototype head and the fixed $r = 0$ control use the same class-mean centers and keep every radius at zero.

Radius and scale sensitivity. For class k, let qp denote the p-th percentile of the training features' distances to their class centroid. Table 12 reports the sensitivity of UMLR^2-S to q50/q80/q90 radius initialization and to $0.1\times/1\times/10\times$ radius learning-rate multipliers, as well as scale initializations $0.5/1/2$. Like the other parameter-matched runs, the sensitivity study uses q80, radius LR factor 1, and scale initialization 1 as its reference setting.

Table 12: Matched radius, radius-learning-rate, and scale initialization sensitivity of UMLR^2-S with frozen ViT-B/16 features at 4 and 16 shots. Accuracies are five-seed means in percent. The last column reports the final mean radius under the aggressive $10\times$ radius learning rate. †Flowers102 has ten training images per class, so this row uses all ten.

Evaluation cell	q50 / q80 / q90 initialization	$0.1\times$ / $1\times$ / $10\times$ radius LR	$0.5/1/2$ scale initialization	Final mean r, $10\times$ LR
CIFAR-100, 4-shot	57.27 / 57.39 / 57.41	57.38 / 57.39 / 57.57	56.69 / 57.39 / 58.22	0.4222
CIFAR-100, 16-shot	67.44 / 67.35 / 67.35	67.34 / 67.35 / 67.67	66.29 / 67.35 / 68.80	0.2956
Flowers102, 4-shot	70.72 / 70.72 / 70.49	70.59 / 70.72 / 71.27	68.66 / 70.72 / 72.91	0.3848
Flowers102, 16-shot†	80.59 / 80.29 / 80.36	80.20 / 80.29 / 81.21	77.75 / 80.29 / 83.59	0.3033
TinyImageNet, 4-shot	70.13 / 70.17 / 70.22	70.15 / 70.17 / 70.39	69.53 / 70.17 / 70.84	0.3365
TinyImageNet, 16-shot	78.43 / 78.45 / 78.46	78.43 / 78.45 / 78.67	77.58 / 78.45 / 79.03	0.2404

All radii stay finite, with final values in $[0.0957, 1.1078]$. Accuracy changes little across q50/q80/q90 initialization and between the $0.1\times$ and $1\times$ radius learning rates in all six cells; the $10\times$ rate shrinks the final mean radius to 0.240–0.422 while accuracy stays at or above the $1\times$ setting. Scale initialization matters more: initialization at 2 improves over 1 in all six cells, by up to 3.30 points, and initialization at 0.5 is consistently lower. In the head-swap and matched experiments, no radius-specific regularizer is used beyond the optimizer's ordinary weight decay; the episodic adaptation in Appendix E adds a weak radius prior.

Parameter-matched recipe. The attribution study (Table 5), the radius-sensitivity study, the radius-only control of Experiment 2, the frozen-versus-fine-tuned comparison, and the balanced full-data and long-tail controls (Appendix F) use a separate parameter-matched recipe. Every head applies an explicit classwise positive scale after its link, so UMLR-H/UMLR^2-H/UMLR-S/UMLR^2-S have the same $K(d+2)$ trainable parameters. Features are train-mean centered and L2-normalized; heads are trained for 60 epochs with AdamW, a cosine schedule, 3 warmup epochs, learning rate 10^{-3}, weight decay 10^{-4}, label smoothing 0.1, and batch size 64. Spherical centers and radii are initialized from train-feature class centroids and the q80 within-class distance, and the effective scale is initialized at 1. Seeds are paired across the four heads within each cell.

D Classwise mixed heads

Mixed-H and Mixed-H^2 use a classwise hard gate between a hyperplane logit and UMLR^2-S:

$$\mu_k^{\mathrm{mix}}(x) = g_k \mu_k^{\mathrm{hyp}}(x) + (1 - g_k)\mu_k^{\mathrm{sph}}(x), \qquad g_k \in \{0, 1\}. \tag{22}$$

Thus $g_k = 1$ selects the hyperplane logit for class k, and $g_k = 0$ selects the spherical logit. The gate is stored as a sigmoid parameter and evaluated with a hard straight-through threshold at 0.5. Its initial hyperplane probability is 0.5. Mixed-H uses $\langle a_k, x\rangle + b_k$ on the hyperplane side; Mixed-H^2 uses $(\langle a_k, x\rangle + b_k)|\langle a_k, x\rangle + b_k|$, without a separate scale. Both use UMLR^2-S on the spherical side, so the two variants differ only in the hyperplane member.

In the low-shot cells, 9,397 of the 9,648 learned class gates (97.4%) select S, and mixing does not consistently improve over the stronger single-geometry head in Table 13. A global choice of geometry therefore suffices in these settings.

Table 13: Exploratory mixed-head suite. Low-shot rows are trained with the mixed-head recipe and full-data rows with the full-data recipe (Table 11). Mixed-H gates UMLR-H with UMLR^2-S; Mixed-H^2 gates UMLR^2-H with UMLR^2-S. Accuracy entries are mean $\pm$ standard deviation over 5 seeds. †Flowers102 has ten training images per class, so its 16-shot and full-data rows use all ten.

Dataset	Backbone	Regime	UMLR-H	UMLR^2-H	UMLR^2-S	Mixed-H	Mixed-H^2
CIFAR-100	ViT-B/16	4-shot	54.26 ± 1.35	55.87 ± 1.35	**57.49 ± 0.44**	54.56 ± 1.18	56.35 ± 0.95
		16-shot	64.89 ± 0.33	68.26 ± 0.46	67.50 ± 0.45	64.91 ± 0.48	**68.41 ± 0.54**
		Full data	88.24 ± 0.42	**88.30 ±0.83**	88.11 ± 0.33	88.26 ± 1.02	88.28 ± 0.28
	ConvNeXt-Tiny	4-shot	44.27 ± 0.58	44.94 ± 0.54	**47.98 ± 0.68**	44.47 ± 0.43	45.57 ± 0.74
		16-shot	57.28 ± 0.60	61.68 ± 0.16	60.82 ± 0.42	57.36 ± 0.74	**61.94 ± 0.21**
		Full data	89.54 ± 0.39	89.79 ± 0.62	89.42 ± 0.77	89.77 ± 0.45	**89.82 ± 0.47**
Flowers102	ViT-B/16	4-shot	62.45 ± 1.00	68.79 ± 0.20	**70.82 ± 0.86**	62.09 ± 1.33	68.76 ± 0.12
		16-shot†	73.33 ± 0.00	**83.01 ± 0.40**	80.65 ± 0.09	73.59 ± 0.37	82.81 ± 0.32
		Full data†	94.31 ± 0.21	94.02 ± 0.42	94.31 ± 0.69	93.82 ± 0.96	**94.61 ± 0.27**
	ConvNeXt-Tiny	4-shot	70.82 ± 0.82	74.54 ± 0.61	**78.27 ± 1.24**	71.47 ± 1.29	74.61 ± 0.92
		16-shot†	81.70 ± 0.23	88.27 ± 0.39	87.35 ± 0.08	81.73 ± 0.32	**88.40 ± 0.23**
		Full data†	95.49 ± 0.39	95.29 ± 0.23	**96.18 ± 0.58**	95.20 ± 0.74	94.71 ± 0.31
TinyImageNet	ViT-B/16	4-shot	67.09 ± 0.74	70.06 ± 0.60	**70.28 ± 0.79**	67.19 ± 0.80	70.27 ± 0.59
		16-shot	77.31 ± 0.51	78.13 ± 0.26	**78.53 ± 0.27**	77.38 ± 0.36	78.16 ± 0.22
		Full data	86.02 ± 0.64	**86.34 ± 0.57**	82.31 ± 0.24	85.40 ± 0.83	86.08 ± 0.76
	ConvNeXt-Tiny	4-shot	53.06 ± 0.36	56.19 ± 0.26	**56.45 ± 0.15**	53.08 ± 0.43	56.34 ± 0.39
		16-shot	67.01 ± 0.08	**70.96 ± 0.18**	69.93 ± 0.08	67.01 ± 0.14	70.89 ± 0.23
		Full data	87.08 ± 0.34	**87.72 ± 0.21**	85.02 ± 0.05	87.14 ± 0.17	87.56 ± 0.19

E Few-shot episodic protocol

Features and adapted heads. All episodic methods use the same ImageNet-pretrained ViT-B/16 features, and all except ProtoNet use CL2N preprocessing: features are centered by the mean feature of the base-class split and L2-normalized. In each episode, the class centers are initialized at the support-class means. The hyperplane heads are initialized at the affine form of the squared-distance prototype score, $\langle 2a_k, x\rangle - \|a_k\|^2$, and the spherical heads at radius r_0, the median over base classes of the q90 within-class distance. All four members are adapted on the support set for 100 AdamW steps with learning rate 0.01, weight decay 10^{-4}, gradient-norm clipping at 10, and an initial effective scale of 10. For UMLR-S and UMLR^2-S, the support loss additionally includes the weak radius prior $10^{-3}\, K^{-1} \sum_k (r_k - r_0)^2$. Neither baseline is adapted: ProtoNet scores queries by negative squared distance to support means of the raw backbone features, and SimpleShot-CL2N applies the same rule to the CL2N features. All methods share the episode draws.

Validation-only member selection. The selection run uses validation classes only, full validation-way episodes, 1/4/16 shots, 15 queries per class, seeds 101–105, and 200 episodes per seed, with the four UMLR members as the only candidates. Each member is scored by its equal-cell mean over the 15 dataset$\times$shot cells, and a single member is selected globally. Members within 0.1 percentage point of the best validation mean would form a practical tie, resolved by the precommitted order UMLR-H, UMLR-S, UMLR^2-H, UMLR^2-S. The validation means are 79.76 (UMLR-H), 78.29 (UMLR^2-H), 80.13 (UMLR-S), and 79.05 (UMLR^2-S), so UMLR-S is selected without a tie. The selection was locked before the test run, which uses disjoint test classes, new seeds 1001–1005, and only ProtoNet, SimpleShot-CL2N, and the selected member (Table 7).

Test-set comparison of all four members. Table 14 evaluates all four members on the test classes with seeds 0–4. Because it gives every member a test-set opportunity, we use it descriptively and not for selection. Against SimpleShot with a ± 0.1-point tie threshold, task-cell W/T/L counts are 20/5/20 for UMLR-H, 7/0/38 for UMLR^2-H, 25/17/3 for UMLR-S, and 15/12/18 for UMLR^2-S. The paired geometry effects $S - H$ are $+0.30, +0.49, +0.14$ points at 1/4/16 shots, whereas $S^2 - H^2$ is $+0.82, -0.13, +1.31$.

Table 14: Test-set comparison of all four UMLR members on the episodic benchmarks. Five-seed mean accuracies (%) for all 15 full-way dataset×shot settings; each row aggregates 1,000 episodes (200 per seed). All methods share features and episode draws.

Dataset	Shot	ProtoNet	SimpleShot-CL2N	UMLR-H	UMLR^2-H	UMLR-S	UMLR^2-S
CIFAR-FS	1	62.76	68.36	67.82	66.90	68.36	68.31
	4	82.04	81.76	81.65	80.29	82.37	79.09
	16	87.10	86.76	88.09	86.69	88.34	87.76
CUB	1	42.31	47.94	47.77	47.40	47.96	47.81
	4	65.44	67.22	71.61	65.83	71.67	68.06
	16	75.67	76.02	84.76	80.55	83.93	83.01
FC100	1	39.32	41.73	41.39	41.03	41.77	41.66
	4	56.62	56.56	56.58	54.68	57.38	54.81
	16	66.53	66.28	67.93	65.57	68.53	67.74
miniImageNet	1	73.57	92.05	92.26	91.51	92.04	92.04
	4	96.71	96.01	96.08	95.73	96.23	94.24
	16	97.71	97.04	97.34	97.30	97.46	97.47
tieredImageNet	1	73.27	82.90	82.31	81.77	82.90	82.90
	4	90.98	90.77	89.78	89.15	90.49	88.83
	16	92.76	92.37	91.88	91.33	92.45	92.01

F Additional controlled experiments

Large-K computation. UMLR-H/UMLR^2-H/UMLR-S/UMLR^2-S each use $K(d+2)$ trainable parameters in the matched implementation. Each class radius changes only its own class logit directly; classes interact through softmax, without an explicit $O(K^2)$ radius interaction. Table 15 reports a controlled A100 benchmark with batch size 64 and $d = 768$. Parameter and measured peak-memory growth are linear in K; the spherical heads have a modest compute/memory overhead from class-center distances.

Table 15: Large-K A100 benchmark. "Time" is the mean forward-plus-backward latency in milliseconds over 50 timed passes after 10 warm-up passes, and peak allocation is in GiB. All four matched heads have the same parameter count at a given K.

		Time (ms)				Peak allocation (GiB)			
K	Params/head	UMLR-H	UMLR^2-H	UMLR-S	UMLR^2-S	UMLR-H	UMLR^2-H	UMLR-S	UMLR^2-S
1,000	0.77M	0.370	0.440	0.659	0.721	0.024	0.024	0.031	0.031
5,000	3.85M	0.373	0.439	0.668	0.721	0.047	0.047	0.079	0.079
10,000	7.70M	0.371	0.439	0.688	0.728	0.078	0.078	0.142	0.142

Frozen versus fine-tuned features. To examine how end-to-end fine-tuning affects the geometry contrasts, we compare the matched four-head family on CIFAR-100, Flowers102, TinyImageNet, Oxford-IIIT Pet [33], and Food-101 [4], the latter evaluated on a seeded 10% hold-out of its official training split, using an ImageNet-pretrained ViT-B/16 that is either frozen or fine-tuned end to end with backbone learning rate 10^{-5}, head learning rate 10^{-3}, and five paired seeds. Table 16 reports all 20 matched dataset–shot–protocol rows. With equal weight per cell, frozen $S - H$ is positive in 7/10 cells (mean $+1.40$ points), whereas fine-tuned $S - H$ is positive in only 1/10 (mean -4.80). For the signed-square link, $S^2 - H^2$ is positive in 6/10 frozen cells (mean -0.04) and 7/10 fine-tuned cells (mean $+0.80$). Under fine-tuning, the hyperplane is preferred with the first-order link and the sphere with the signed-square link in most cells, so the preferred geometry depends on both the link and the training protocol.

Table 16: Frozen versus end-to-end fine-tuned ImageNet-pretrained ViT-B/16 backbones under the parameter-matched recipe. UMLR head entries are five-seed mean ± sample standard deviation accuracies (%); effect columns use the compact $S - H$ notation and report paired mean differences in percentage points. †Flowers102 has ten training images per class, so these rows use all ten.

Dataset/regime	Protocol	UMLR-H	UMLR^2-H	UMLR-S	UMLR^2-S	$S - H$	$S^2 - H^2$
CIFAR-100, 4-shot	Frozen	54.27 ± 1.73	56.12 ± 1.60	56.03 ± 0.19	57.39 ± 0.49	$+1.76$	$+1.27$
CIFAR-100, 4-shot	Fine-tuned	59.22 ± 0.80	61.28 ± 0.90	58.02 ± 0.72	64.33 ± 0.67	-1.20	$+3.04$
CIFAR-100, 16-shot	Frozen	66.64 ± 0.69	68.27 ± 0.59	65.68 ± 0.75	67.35 ± 0.54	-0.96	-0.92
CIFAR-100, 16-shot	Fine-tuned	75.06 ± 0.30	76.18 ± 0.19	66.97 ± 0.08	76.65 ± 0.20	-8.09	$+0.48$
Flowers102, 4-shot	Frozen	62.91 ± 1.09	69.15 ± 0.32	68.59 ± 1.25	70.72 ± 1.06	$+5.69$	$+1.57$
Flowers102, 4-shot	Fine-tuned	80.72 ± 0.82	83.33 ± 0.64	73.95 ± 1.08	83.30 ± 0.91	-6.76	-0.03
Flowers102, 16-shot†	Frozen	75.36 ± 0.30	83.30 ± 0.54	76.21 ± 0.32	80.29 ± 0.10	$+0.85$	-3.01
Flowers102, 16-shot†	Fine-tuned	92.55 ± 0.26	92.81 ± 0.34	81.99 ± 0.54	93.95 ± 0.28	-10.56	$+1.14$
TinyImageNet, 4-shot	Frozen	67.27 ± 0.92	70.14 ± 0.67	69.48 ± 0.69	70.17 ± 0.97	$+2.21$	$+0.03$
TinyImageNet, 4-shot	Fine-tuned	70.38 ± 0.21	73.86 ± 0.76	68.94 ± 0.42	75.79 ± 0.53	-1.44	$+1.94$
TinyImageNet, 16-shot	Frozen	78.82 ± 0.27	78.02 ± 0.19	76.50 ± 0.27	78.45 ± 0.36	-2.32	$+0.43$
TinyImageNet, 16-shot	Fine-tuned	82.03 ± 0.19	84.67 ± 0.02	75.31 ± 0.16	83.99 ± 0.16	-6.72	-0.68
Oxford-IIIT Pet, 4-shot	Frozen	83.18 ± 0.73	83.47 ± 0.79	86.84 ± 0.33	86.35 ± 0.64	$+3.66$	$+2.88$
Oxford-IIIT Pet, 4-shot	Fine-tuned	86.14 ± 0.25	88.13 ± 0.39	87.93 ± 0.38	88.04 ± 0.68	$+1.79$	-0.08
Oxford-IIIT Pet, 16-shot	Frozen	88.46 ± 0.39	90.62 ± 0.64	89.70 ± 1.08	89.92 ± 0.77	$+1.24$	-0.70
Oxford-IIIT Pet, 16-shot	Fine-tuned	91.53 ± 0.30	91.85 ± 0.29	90.48 ± 0.43	92.01 ± 0.19	-1.05	$+0.16$
Food-101, 4-shot	Frozen	29.87 ± 1.99	33.16 ± 1.34	33.87 ± 1.58	34.20 ± 1.30	$+4.00$	$+1.04$
Food-101, 4-shot	Fine-tuned	35.98 ± 1.99	37.05 ± 1.50	33.61 ± 1.42	38.83 ± 1.16	-2.37	$+1.78$
Food-101, 16-shot	Frozen	45.27 ± 0.33	49.18 ± 0.31	43.18 ± 0.47	46.17 ± 0.44	-2.09	-3.01
Food-101, 16-shot	Fine-tuned	52.08 ± 0.43	53.38 ± 0.22	40.45 ± 0.30	53.67 ± 0.48	-11.63	$+0.29$

Balanced full-data and long-tailed training. We also test whether the low-shot behavior persists under balanced full-data training and class imbalance. These runs use the parameter-matched recipe with frozen backbones for 60 epochs, unlike the fine-tuned full-data rows of Table 4, which explains their lower absolute accuracies. Table 17 reports five-seed means. All balanced full-data geometry contrasts are negative for all five paired seeds, and the 95% confidence intervals exclude zero. On CIFAR-100-LT, built with the exponential class profile of Cao et al. [6], Cui et al. [14], the spherical members fall further behind the hyperplane members than under balanced training. At imbalance factor 0.1 (the rarest class has one tenth as many training images as the most frequent; 0.01 means one hundredth), the paired effects are $S - H = -40.52 \pm 0.23$ points (95% t-CI $[-41.08, -39.96]$) and $S^2 - H^2 = -2.02 \pm 0.09$ points ($[-2.24, -1.79]$); at factor 0.01, they are -33.27 ± 0.37 points ($[-34.20, -32.34]$) and -3.64 ± 0.27 points ($[-4.31, -2.97]$), respectively. For first-order UMLR-S, the learned-radius/ class-frequency Spearman correlations are 0.937 and 0.631 at imbalance factors 0.1 and 0.01, while tail-class accuracies are only 1.95% and 0.01%. In these balanced full-data and long-tailed regimes, the hyperplane head is therefore preferred.

Table 17: Balanced full-data and CIFAR-100 long-tail controls with frozen backbones under the parameter-matched recipe. Head entries are five-seed mean accuracies (%); effect columns are paired mean differences in percentage points.

Dataset/regime	Backbone	UMLR-H	UMLR^2-H	UMLR-S	UMLR^2-S	$S - H$	$S^2 - H^2$
CIFAR-100, balanced full	ViT-B/16	81.58	81.05	68.64	79.41	-12.94	-1.64
CIFAR-100, balanced full	ConvNeXt-Tiny	79.61	78.99	61.36	76.52	-18.24	-2.47
CIFAR-100-LT, 0.1	ViT-B/16	76.96	76.01	36.44	73.99	-40.52	-2.02
CIFAR-100-LT, 0.01	ViT-B/16	65.54	65.44	32.27	61.80	-33.27	-3.64

G Assets and licenses

All datasets are public research benchmarks, used under their creators' terms: CIFAR-100 [21] and the CIFAR-derived CIFAR-FS [2] and FC100 [32]; Flowers102 [30]; Oxford-IIIT Pet [33], released under CC BY-SA 4.0; Food-101 [4]; CUB-200-2011 [43]; and ImageNet-1K [15] together with the ImageNet-derived TinyImageNet [22], miniImageNet [42], and tieredImageNet [34], which follow the ImageNet terms of access for non-commercial research. The ImageNet-pretrained ViT-B/16 [18] and ConvNeXt-Tiny [26] weights are obtained from torchvision, which is distributed under the BSD-3-Clause license. No new dataset is released.

H Proofs

Hyperplane. For the hyperplane case in Eq. (1), the defining function is $u_{a,b}(x) = \langle a, x\rangle + b$, and $\mathcal{U}_{a,b} = \{x \mid u_{a,b}(x) = 0\}$. Euclidean geometry gives

$$\mathrm{d}\left(x, \mathcal{U}_{a,b}\right) = \frac{|u_{a,b}(x)|}{\|a\|}. \tag{23}$$

Substituting this distance into Eq. (4) gives

$$\operatorname{sign}(u_{a,b}(x))\, \|a\|\, \mathrm{d}\left(x, \mathcal{U}_{a,b}\right) = u_{a,b}(x), \tag{24}$$

which is the UMLR-H expression in Eq. (5). Substituting the same distance into Eq. (6) gives

$$\operatorname{sign}(u_{a,b}(x))\, \|a\|\, \mathrm{d}\left(x, \mathcal{U}_{a,b}\right)^2 = \frac{1}{\|a\|} \operatorname{sign}(u_{a,b}(x)) u_{a,b}(x)^2, \tag{25}$$

which is the UMLR^2-H expression in Eq. (7).

Sphere. For the spherical case in Eq. (1), the defining function is $u_{a,b}(x) = b - \|x - a\|$, and $\mathcal{U}_{a,b} = \{x \mid u_{a,b}(x) = 0\}$. The distance to the sphere is

$$\mathrm{d}\left(x, \mathcal{U}_{a,b}\right) = |b - \|x - a\|\,| = |u_{a,b}(x)|. \tag{26}$$

Substituting this distance into Eq. (4) gives

$$\operatorname{sign}(u_{a,b}(x))\, \|a\|\, \mathrm{d}\left(x, \mathcal{U}_{a,b}\right) = \|a\|\, (b - \|x - a\|)\,, \tag{27}$$

which is the UMLR-S expression in Eq. (5). The squared-distance substitution gives

$$\operatorname{sign}(u_{a,b}(x))\, \|a\|\, \mathrm{d}\left(x, \mathcal{U}_{a,b}\right)^2 = \|a\| \operatorname{sign}(b - \|x - a\|)\, (b - \|x - a\|)^2\,, \tag{28}$$

which is the UMLR^2-S expression in Eq. (7). In the implemented parameterization, the geometric factors, $\|a\|$ for the sphere and $\|a\|^{-1}$ for UMLR^2-H, are replaced by a class scale α_k, learned or fixed at 1 depending on the recipe (Section 4).

Prototype as the zero-radius limit. The implemented UMLR^2-S logit is $\alpha_k(b_k - \|x - a_k\|)|b_k - \|x - a_k\||$. Setting $b_k = 0$ and $\alpha_k = 1$ gives

$$(-\|x - a_k\|)|-\|x - a_k\|| = -\|x - a_k\|^2\,, \tag{29}$$

which is the squared-distance prototype score. Expanding this score gives

$$-\|x - a_k\|^2 = 2\,\langle a_k, x\rangle - \|a_k\|^2 - \|x\|^2\,. \tag{30}$$

The last term is shared across classes, so it cancels in the softmax and leaves the class probabilities unchanged. The remaining score $\langle 2a_k, x\rangle - \|a_k\|^2$ is an MLR logit whose bias is tied to its weight, the algebraic identity between squared prototype scoring and MLR noted by Snell et al. [38].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction present UMLR as a family that contains MLR as its hyperplane member and the squared-distance prototype classifier as the zero-radius limit of its spherical member, and they summarize the matched experiments, which find the hyperplane and spherical geometries complementary across data regimes and score links.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

Justification: Section 7 discusses the head-level scope, the protocol dependence revealed by fine-tuning, the full-data and long-tail regimes in which the hyperplane head is preferred, scale-initialization sensitivity, and the untested modalities and backbones.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete proof?

   Answer: [Yes].

   Justification: The main mathematical claims are elementary signed-distance identities and a classical geometric classification result, cited to O'Neill [31]. Assumptions are stated in the corresponding statements, and derivations are given in Appendix H.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and conclusions of the paper?

   Answer: [Yes].

   Justification: Appendices C and E specify the datasets, splits, backbones, feature preprocessing, heads, initialization, optimization settings, and evaluation protocols, and the released code provides one script per experiment with the exact arguments.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results?

   Answer: [Yes].

   Justification: The code is released at `https://github.com/NickyoyoSu/Umbilic-Multinomial-Logistic-Regression` under the MIT license, with installation and data-preparation instructions and one script per experiment. All datasets are public benchmarks.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [Yes].

   Justification: The main text specifies the core comparisons. Appendix C lists the datasets, backbones, feature regimes, heads, sampling protocol, and optimization settings for each experiment family, and Appendix E specifies the episodic adaptation and validation-selection protocol.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about statistical significance?

   Answer: [Yes].

   Justification: Tables report mean $\pm$ standard deviation (population standard deviation for the low-shot rows of Tables 4 and 13, sample standard deviation elsewhere) over five independent training seeds, and paired geometry effects are reported with 95% hierarchical-bootstrap or t confidence intervals over paired seeds.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the compute resources needed to reproduce the experiments?

   Answer: [Yes].

   Justification: All runs used the Leonardo supercomputer (CINECA). Each training run used one NVIDIA A100 (64 GB) GPU, 8 CPU cores, and 64 GB of host memory; a frozen-feature head run takes about 5–30 minutes and an end-to-end fine-tuning run a few hours. The one-time feature caches used four A100 GPUs and 240 GB of host memory for ImageNet-1K and 256 GB of host memory for tieredImageNet; the ImageNet-1K and episodic heads then

train on cached features. The full project, including preliminary and discarded runs, used more compute than the reported experiments.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes].

   Justification: The work uses public benchmarks and studies classification-head geometry. No human-subject data collection, private data, or deployment study is involved.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Section 7 includes a broader-impact statement. The work analyzes classifier-head geometry on public image benchmarks and does not introduce a deployed system, dataset, or generative model.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A].

    Justification: The work does not release high-risk generative models, private datasets, or models intended for deployment.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited and are license and terms of use explicitly mentioned?

    Answer: [Yes].

    Justification: All datasets and pretrained models are cited, and Appendix G states their licenses or terms of use.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The only new asset is the released code, which is documented in its README and licensed under MIT. No new dataset is introduced.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and details about compensation?

    Answer: [N/A].

    Justification: The work does not involve crowdsourcing or human-subject experiments.

15. **Institutional review board approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants and whether IRB approvals or equivalent approvals were obtained?

    Answer: [N/A].

    Justification: The work does not involve human-subject experiments.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [N/A].

    Justification: LLMs are not a component of the proposed method or experimental system. Any writing assistance does not affect the scientific method or results.