Orthogonal Origin Parking: Decoupling Lorentz Manifolds for Robust OOD Generalization

Peter J.T. Kampen[1] **Anders Nymark Christensen**[1] **Morten Rieger Hannemose**[1]
Anders Dahl[1] **Josefine Vilsbøll Sundgaard**[1,2]
[1]DTU Compute, Technical University of Denmark [2]Novo A/S, Denmark
{pjtka,anym,mohan,abda,josh}@dtu.dk, JFVS@novonordisk.com

Abstract

Fine-grained classification in deep taxonomies suffers from representational crowding: large macro-classes dominate the feature space, leaving little angular capacity for fine-grained distinctions, particularly under out-of-distribution (OOD) shifts. Standard hyperbolic spaces offer exponential volume for tree-like data, but embedding an entire taxonomy into a single manifold forces distinct macro-branches to compete for shared angular capacity and entangles their optimization updates. We introduce a geometric architecture that embeds hierarchical taxonomies into a Cartesian product of Lorentz manifolds. Our core mechanism, *Orthogonal Origin Parking* (OOPark), decouples the taxonomy at the root level by assigning an independent Lorentz manifold to each macro-branch and penalizing inactive embeddings that deviate from the manifold origin. We evaluate across four biological OOD benchmarks (skin lesions, iWildCam, fungi, and plankton) characterized by hierarchical class structure, class imbalance, and domain shift. OOPark generally outperforms standard Euclidean and single-manifold hyperbolic baselines while using 40–100-dimensional product embeddings (20 dimensions per macro-branch) against Euclidean baselines of up to 512 dimensions, preserving both fine-grained accuracy and global taxonomic fidelity under severe distribution shifts. Code is available at `https://github.com/PeterKampen/OOPark`.

1 Introduction

Although hyperbolic embeddings naturally encode hierarchical structures, classification performance degrades when macro-branches must compete for finite angular bandwidth. Overcrowding by dominant branches compresses the representational space, degrading fine-grained separation. We show that this geometric bottleneck can be mitigated by decoupling branches at the root and embedding them into independent sub-manifolds. Our Orthogonal Origin Parking (OOPark) method implements this structural isolation explicitly: the active semantic branch is freely embedded, while all inactive branches are pulled towards the origin of their respective hyperbolic spaces. The result is improved out-of-distribution (OOD) classification performance while keeping the overall embedding dimensionality low.

In high-risk domains, the severity of a classification error is often dictated by its semantic distance within a taxonomy [3, 10]. Architectures lacking explicit structural priors frequently suffer from hierarchical collapse under OOD shifts, grouping visually similar but disparate classes. By embedding the taxonomy directly into the representation space, classifiers preserve global structural fidelity, which facilitates graceful degradation [50].

Standard networks embed features into flat Euclidean spaces ($\mathbb{R}^N$), which distort tree-like structures whose node counts grow exponentially with depth [26, 42]. This forces deep taxonomic branches to crowd, entangling features that should remain structurally distinct [9, 34, 42]. To overcome this,

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

hyperbolic space has emerged as a natural framework for representation learning of tree-structured data [37]. Characterized by constant negative curvature, hyperbolic geometry provides exponential volume expansion, enabling representations that preserve ancestral relationships and minimize semantic entanglement [34].

Deep networks based on the standard Poincaré ball suffer from numerical instability, boundary collapse, and vanishing gradients during optimization [25, 34]. Instead, modern architectures embed features directly onto the Lorentz manifold (the hyperboloid model) [33, 41] whose unbounded domain avoids boundary collapse and for which the ambient Minkowski inner product yields a numerically robust distance [6, 25, 33]. To enforce classification within this geometry, prototype learning offers an efficient alternative to pair-wise metric paradigms [45]. By mapping inputs directly to target centroids, this approach significantly stabilizes optimization [32] and regularizes the embedding space [52].

However, embedding an entire taxonomy into a single hyperbolic manifold introduces a capacity bottleneck [13, 42], which we trace to three compounding properties. (i) Because all continuous hyperbolic spaces expand from a single origin, independent semantic branches partition a shared, finite angular bandwidth [13]. (ii) Standard margin-based objectives disproportionately allocate this limited angular capacity to macro-level separation [5, 20]. Consequently, the representational volume available for deep, fine-grained sub-trees is restricted, inducing a structural pathology we term *angular crowding*.

(iii) Angular crowding becomes particularly pronounced under OOD conditions. Real-world distribution shifts induce internal covariate shifts, perturbing latent features away from their source distributions [17, 22]; in hyperbolic space, such features also contract toward the origin, where angular margins shrink further [20] (Section 3). Because fine-grained classes occupy narrow angular margins, slight perturbations push representations across decision boundaries [8, 24]. Consequently, the shifted features miss their designated angular segments and cross into taxonomically disparate branches.

To resolve angular crowding, we introduce a Cartesian product of independent Lorentz manifolds in which the taxonomy acts as a structural prior on the embedding geometry rather than as an auxiliary loss. The taxonomy enters only through where the prototypes are placed in this product space, with one sub-manifold per macro-branch, and upper-level nodes are discarded once placement is complete. Like the Euclidean baselines, our network is trained on flat leaf-node prediction; samples are projected into all sub-manifolds, but unconstrained optimization across irrelevant manifolds generates representational interference. We enforce decoupling via *Orthogonal Origin Parking* (OOPark): a prototype-level regularization that constrains embeddings in inactive sub-manifolds to remain near fixed origin prototypes. This isolates spatial variance to the active sub-manifold, stabilizes optimization, and improves OOD robustness. We evaluate OOPark on four OOD benchmarks (skin lesions, iWildCam, fungi, and plankton) characterized by deep taxonomic structure, class imbalance, and domain shift. In summary, our contributions are:

- **Hyperbolic Product Spaces with Origin Parking:** We partition the embedding space into independent Lorentz sub-manifolds, one per macro-branch, and anchor inactive branches to locked origin prototypes, isolating the optimization of disparate macro-taxonomies.
- **Improved OOD Generalization:** We demonstrate that our architecture mitigates feature collapse under severe distribution shifts. It outperforms baselines by making "taxonomically safe" errors, routing ambiguous OOD samples toward their correct ancestral lineage.
- **Improved Capacity Allocation:** The orthogonal decoupling mitigates the angular bottleneck of single-origin geometries, preserving complex hierarchies with 40–100-dimensional product embeddings (20 dimensions per macro-branch) against Euclidean baselines of up to 512 dimensions.

2 Related work

Recent metric learning approaches have transitioned from Euclidean centroids [45] to fixed hyperbolic prototypes [12, 20]. To maximize spatial flexibility, modern architectures frequently optimize manifold curvature as a learnable parameter [5] and utilize multi-dimensional scaling (MDS) [23] to encode prior hierarchical structures [42]. However, these formulations embed the entire taxonomy

into a single continuous hyperbolic manifold, enforcing a uniform global curvature that assumes structural homogeneity across the tree topology. Consequently, they remain bounded by the finite angular bandwidth of a single origin, a fundamental capacity bottleneck that limits the representation of complex, heterogeneous hierarchies [13, 42]. By forcing disparate semantic branches to compete for shared geometric capacity, these single-manifold architectures are exposed to angular crowding (Section 5). Beyond prototype-based classification, hyperbolic geometry has also been deployed for hierarchical reasoning via hyperbolic graph neural networks [5, 27], which propagate taxonomic structure through manifold-aware message passing rather than through fixed centroids. Other hierarchy-aware methods inject the taxonomy into the supervision through hierarchical losses and soft labels [3, 50]. Closer to ours, metric-guided prototype learning (MGP) [10] keeps flat supervision and arranges learned Euclidean prototypes to follow the hierarchy; we include it as a baseline (Section 5). OOPark likewise uses the taxonomy only to place prototypes, but in a sparse product of hyperbolic sub-manifolds.

To embed heterogeneous topologies, Cartesian product spaces combining spherical, Euclidean, and hyperbolic manifolds have been introduced [13, 44]. This mixed-curvature paradigm remains highly active in modern geometric deep learning, utilizing product spaces to embed heterogeneous local graph topologies [1], learn self-supervised graph representations [46] and dynamically route representations through a mixture of Riemannian experts [15]. However, these methods primarily utilize product spaces to capture mixed structural signatures, fitting each factor's curvature to local topology rather than a fixed taxonomy, and can lead to feature drift during unconstrained optimization. Our architecture fundamentally repurposes this framework: rather than mixing geometries, we deploy a Cartesian product of negative-curvature Lorentz sub-manifolds as an explicit hierarchical routing mechanism. By anchoring inactive sub-spaces via Origin Parking, we address both the unconstrained drift of prior product models, isolated by our unparked product ablation (Section 5), and the representational crowding of single-manifold baselines. Domain-generalization methods [22] and hyperspherical OOD detectors [31] act at the feature level, whereas OOPark structures the output space and its classification head, so they are complementary and could be combined.

Preliminaries. Hyperbolic space admits several isometric models, including the Poincaré ball, the Klein model and the Lorentz (hyperboloid) model [41], so angular crowding does not depend on the choice of model; we use the Lorentz model [25, 33].

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{l+1}$. The Minkowski inner product is defined by:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^{l} x_i y_i. \tag{1}$$

For vectors with $\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} > 0$, such as the tangent vectors used below, the Minkowski inner product induces the norm $||\mathbf{x}||_{\mathcal{L}} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}}}$. The Lorentz manifold $\mathbb{L}_k^l$ is represented as the upper sheet of a hyperboloid in $l+1$ dimensional space, characterized by a learnable curvature scale $k > 0$ (sectional curvature $-1/k$):

$$\mathbb{L}_k^l = \left\{ \mathbf{x} \in \mathbb{R}^{l+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -k, x_0 > 0 \right\}. \tag{2}$$

For any two points $\mathbf{x}, \mathbf{y} \in \mathbb{L}_k^l$, the geodesic distance is induced by the Minkowski inner product and defined as:

$$d_{\mathbb{L}}(\mathbf{x}, \mathbf{y}) = \sqrt{k} \operatorname{arcosh}\left(-\frac{\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}}{k} \right). \tag{3}$$

Features extracted by the standard Euclidean backbone (e.g., ConvNeXt) fundamentally reside in flat space. We define this flat Euclidean output space as the tangent space at the origin of the Lorentz manifold, denoted as $\mathcal{T}_{\mathbf{0}}\mathbb{L}_k^l$. In the Lorentz model, the origin is defined as $\mathbf{0}_{\mathbb{L},k} = (\sqrt{k}, 0, \ldots, 0) \in \mathbb{R}^{l+1}$. The tangent space at this origin is the orthogonal complement with respect to the Minkowski inner product [6]:

$$\mathcal{T}_{\mathbf{0}}\mathbb{L}_k^l = \left\{ \mathbf{v} \in \mathbb{R}^{l+1} : \langle \mathbf{v}, \mathbf{0}_{\mathbb{L},k} \rangle_{\mathcal{L}} = 0 \right\} \Rightarrow \mathbf{v} = (0, v_1, \ldots, v_l). \tag{4}$$

This construction naturally provides an isometric mapping to $\mathbb{R}^l$, allowing the network's Euclidean features to be smoothly integrated.

To project these Euclidean features from the tangent space onto the hyperbolic manifold, we utilize the Exponential Map $\exp_{\mathbf{0}} : \mathcal{T}_{\mathbf{0}}\mathbb{L}_k^l \to \mathbb{L}_k^l$, which preserves the distance from the origin. For any tangent vector $\mathbf{v} \in \mathcal{T}_{\mathbf{0}}\mathbb{L}_k^l$ where $\mathbf{v} \neq \mathbf{0}$, the projection is formalized as [9, 33]:

$$\exp_{\mathbf{0}}(\mathbf{v}) = \cosh\left(\frac{||\mathbf{v}||_{\mathcal{L}}}{\sqrt{k}}\right)\mathbf{0}_{\mathbb{L},k} + \sqrt{k}\sinh\left(\frac{||\mathbf{v}||_{\mathcal{L}}}{\sqrt{k}}\right)\frac{\mathbf{v}}{||\mathbf{v}||_{\mathcal{L}}}. \tag{5}$$

This mapping ensures that the distance from the origin is preserved.

3 Method

Angular Crowding in a Single Manifold. In continuous hyperbolic space, the radial distance from the origin correlates with predictive certainty [33, 20]. For a predicted embedding at radius r and boundary prototypes for a minority class $\mathbf{y}$, the spatial margin spanning $\mathbf{y}$'s angular bandwidth $\Delta\theta_y$ is the arc length $\sqrt{k}\sinh(r/\sqrt{k})\,\Delta\theta_y$. OOD samples lack discriminative features, causing representations to collapse toward the locally Euclidean origin ($r \to 0$) [33, 20]; in this limit the margin reduces to $r\Delta\theta_y$. Since frequency-weighted capacity allocation assigns narrow angular measure to minority classes, decreasing r directly shrinks this boundary, causing slight representational variance to push uncertain inputs across into an adjacent macro-branch (Fig. 3). Because the radius–certainty link is itself an empirical regularity [20], this and the arguments below are analytical intuitions that motivate the design rather than formal guarantees; we test them empirically in Section 5.

To mitigate this problem of angular crowding, we decouple the embedding space into a Cartesian product of N independent Lorentz sub-manifolds, where each sub-manifold corresponds to a distinct Level-1 (L1) macro-branch (e.g., Phylum):

$$\mathcal{M} = \prod_{m=1}^{N} \mathbb{L}_{k_m}^l, \tag{6}$$

where l and k_m denote the dimensionality and learnable curvature of the m-th sub-manifold.

Rather than relying on continuous angular boundaries to separate macro-branches, we introduce an active routing mechanism termed **Orthogonal Origin Parking** (OOPark). Let $\mathbf{c}_i \in \mathcal{M}$ represent the locked, pre-computed prototype for class i, which belongs to the L1 macro-branch j; a prototype is a fixed target coordinate to which samples of its class are mapped, and classification is by nearest prototype [45]. We define the prototype via its component $\mathbf{c}_{i,m}$ within the m-th sub-manifold:

$$\mathbf{c}_{i,m} = \begin{cases} \mathbf{x}_i \in \mathbb{L}_{k_j}^l & \text{if } m = j \quad \text{(Active Branch)} \\ \mathbf{0}_m \in \mathbb{L}_{k_m}^l & \text{if } m \neq j \quad \text{(Inactive / Parked)} \end{cases}, \tag{7}$$

where $\mathbf{0}_m = (\sqrt{k_m}, 0, \ldots, 0) \in \mathbb{R}^{l+1}$ is the geometric origin of the sub-manifold, and $\mathbf{x}_i$ is the pre-allocated target coordinate. We call the inactive components $\mathbf{c}_{i,m}$, $m \neq j$, *parked*, and refer to the product space together with this parking scheme as the *Parked Product Space*. The squared geodesic distance between the prototype and the m-th component $\mathbf{p}_m \in \mathbb{L}_{k_m}^l$ of a predicted composite embedding is:

$$d^2_{\mathbb{L}_m}(\mathbf{c}_{i,m}, \mathbf{p}_m) = k_m \operatorname{arcosh}^2\left(-\frac{\langle \mathbf{c}_{i,m}, \mathbf{p}_m \rangle_{\mathcal{L}_m}}{k_m}\right). \tag{8}$$

Asymmetric Distance Calculation. Following prototype learning [45], we optimize the visual encoder via cross-entropy over a probability distribution across predefined classes. Computing the standard composite Pythagorean distance across all sub-manifolds introduces a counterproductive gradient: since all parked prototypes share the exact same mathematical origin, including parked components in the negative-class denominator would generate gradients that push embeddings *away* from the origin, even though this origin must act as a null sink for inactive components. To preserve the null sink, we use an asymmetric distance. During optimization, the distance to the active prototype component is computed for *all* classes; the orthogonal parking penalty (distance to the inactive origins) is enforced only for the ground-truth class y. The effective squared distance $\tilde{D}_i$ to any class i in L1

branch $b(i)$ is:

$$\begin{aligned}\tilde{D}_i &= d^2_{\mathbb{L}_{b(i)}}(\mathbf{p}_{b(i)}, \mathbf{c}_{i,b(i)}) + \mathbb{I}[i = y] \sum_{m \neq b(y)} d^2_{\mathbb{L}_m}(\mathbf{p}_m, \mathbf{c}_{y,m}) \\ &= d^2_{\mathbb{L}_{b(i)}}(\mathbf{p}_{b(i)}, \mathbf{x}_i) + \mathbb{I}[i = y] \sum_{m \neq b(y)} d^2_{\mathbb{L}_m}(\mathbf{p}_m, \mathbf{0}_m),\end{aligned} \tag{9}$$

where $\mathbb{I}[\cdot]$ is the indicator function. We optimize the network using cross-entropy [51] $\mathcal{L}_{\text{CE}} = -\log \frac{\exp(-\tilde{D}_y)}{\sum_{i=1}^{|C|} \exp(-\tilde{D}_i)}$. This asymmetric formulation ensures that active components undergo standard spatial competition in the denominator, while inactive components are pulled to the origin exclusively via the loss numerator, effectively shielding parked branches from contrastive spatial repulsion. Because all classes share one softmax, this competition also spans macro-branches; separately trained per-branch models, each seeing only its own sub-tree, would receive no cross-branch negatives and could not route a sample out of the wrong branch. By the definition of the exponential map at the origin [33], the asymmetric distance bounds the L_2 norm of the Euclidean tangent vectors ($||\mathbf{v}_m||_2$) for inactive branches. This avoids the premature optimization trade-offs of global weight regularization [20, 51], leaving the active projection head spatially unconstrained. Simultaneously, to minimize the loss, the shared backbone is compelled to route the features of any given class into the null space of all irrelevant projection heads ($W_{m \neq y}\mathbf{x} \approx 0$), organically forcing semantic information into orthogonal sub-spaces. Furthermore, this joint objective protects the learnable curvature parameters (k_m) from degenerate collapse. The origin penalty exerts downward pressure on k_m; any artificial flattening simultaneously shrinks the spatial margin to the negative classes in those manifolds, inflating the contrastive denominator and creating a *geometric equilibrium, a stable balance between the parking and contrastive terms (Fig. 6),* that ensures k_m reflects true macro-taxonomic variance. To prevent boundary drift [20, 33], we apply an exponential radius penalty $\mathcal{L}_{\text{rad}} = \alpha \exp\left(d_{\mathbb{L}_{b(y)}}(\mathbf{p}_{b(y)}, \mathbf{0}_{b(y)}) - R_{\text{max}}\right)$, a smooth penalty that is small but non-zero near the origin and grows exponentially as the radius approaches R_{max}. OOPark does not benefit from it (Fig. 2a); we retain it because the single-manifold and unparked baselines do, so applying it uniformly keeps the comparison fair. The network is jointly optimized via $\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{rad}}$.

Origin Parking does not discard the fine-grained taxonomy; it protects it. Within each active j-th sub-manifold ($\mathbb{L}^l_{k_j}$), we apply hyperbolic multi-dimensional scaling (h-MDS; see Section E) [42], a standard embedding method that we run with a single configuration shared across all benchmarks, over the full sub-tree (from L2 to the leaf level): pairwise distances between all taxonomic nodes define the h-MDS objective and shape the resulting geometry. After convergence, only the leaf-node coordinates are retained as classification prototypes. The classification loss, therefore, reduces to a flat softmax over leaf prototypes, while the deeper hierarchy persists as a geometric prior on prototype placement. By neutralizing the extreme variance of OOD samples at the macro-level, the active sub-manifold focuses on fine-grained intra-branch differentiation. During inference, the ground-truth label y is unknown, so the asymmetric training distance is inapplicable. We score each candidate class using only the geodesic distance in the candidate's own active sub-manifold, ignoring the parked-prototype components in every other sub-manifold. Classification of an unseen query $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_N)$ proceeds by

$$y^* = \arg\min_{i \in \mathcal{C}} \; d^2_{\mathbb{L}_{b(i)}}\left(\mathbf{q}_{b(i)}, \mathbf{c}_{i,b(i)}\right), \tag{10}$$

where $b(i)$ is the L1 macro-branch of class i and $\mathbf{c}_{i,b(i)} \in \mathbb{L}^l_{k_{b(i)}}$ is its active prototype component, defined via Eq. 7. Macro-routing is implicit in this rule: the encoder drives query projections in inactive sub-manifolds toward the origin, so a wrong-branch candidate evaluates against $\mathbf{q}_{b(i)} \approx \mathbf{0}_{b(i)}$ while its prototype lies away from the origin, yielding systematically large distances; only candidates from the correct branch compete on fine-grained intra-branch geometry. The heterogeneous learnable curvatures (k_m) implicitly weight each sub-manifold's intra-branch metric. All candidates within a branch share the same parked components, so these components cannot change the ranking within a branch; the inference rule therefore stays consistent with the asymmetric training distance, which likewise keeps parked components out of the competition between classes.

Table 1: Dataset statistics summarizing hierarchical scale and distribution shifts. To reflect structural differences between the source and target domains, all metrics are reported as (Train, OOD) tuples. $|C_{L1}|$ and $|C_{\text{Leaf}}|$ denote the number of macro and fine-grained classes, respectively, while N denotes the total number of images. We refer to the task by **Name**.

Name	**Datasets (Train, OOD)**	**Depth**	$\lvert C_{L1}\rvert$	$\lvert C_{\text{Leaf}}\rvert$	N
Plankton	(WHOI, FlowCamNet)	(5, 5)	(4, 3)	(44, 10)	(263K, 52K)
Fungi	(DK Fungi, Fungi-tastic)	(6, 6)	(5, 3)	(1536 , 1306)	(264K, 8.5K)
iWildCam	(iWildCam ID, iWildCam OOD)	(5, 5)	(3, 2)	(86, 51)	(68K, 14K)
Skin lesion	(AISC, ISIC 2019 Train)	(3, 2)	(2, 2)	(45, 8)	(37K, 25K)

4 Implementation & Experiments

Few image benchmarks combine a deep taxonomy with a genuine distribution shift; with ImageNet-pretrained backbones, non-biological categories largely fall inside the pretraining distribution, making such a shift hard to construct. We therefore construct four biological distribution-shift evaluations spanning seven datasets and three imaging modalities (dermoscopy, camera-trap and field photographs, and imaging flow cytometry). Hierarchical depths across these domains range from $L = 3$ to $L = 6$; training and OOD taxonomies have equal depth except for skin lesions, and OOD sets contain fewer classes and images because not every class occurs in the target domain. Dataset statistics are presented in Table 1.

Skin lesions (AISC & ISIC 2019) We train on the AISC dataset [48] and evaluate on the ISIC 2019 training set [7, 49] previously established as a severe distribution shift [48]. While AISC contains labels at both Layer 2 and Layer 3, ISIC 2019 is restricted to Layer 2, and "leaf" performance reflects Layer 2 performance. While we train on the proprietary AISC dataset, evaluation is performed on the publicly available ISIC dataset.

iWildCam (iWildCam v2.0) We utilize this dataset from the WILDS benchmark [2, 22], which provides explicitly curated train/validation/test splits to evaluate OOD generalization across disparate camera deployments. We adopt the canonical WILDS `ood_val` split [22] as our validation set.

Fungi (Danish Fungi & Fungi-tastic) Models are trained on the Danish Fungi dataset [39] and evaluated on FungiTastic [38] restricted to its non-Danish subset, enforcing a strict geographic distribution shift.

Plankton (WHOI & FlowCamNet) We establish a marine distribution shift by training on the WHOI Plankton dataset [35] and evaluating on FlowCamNet [19]. This cross-oceanic, cross-instrument setup introduces three distinct distribution shifts: (1) differing optical physics between IFCB and FlowCAM images, (2) phenotypic variance across distinct marine ecosystems, and (3) drastically altered ecological class priors.[1]

Except for the Skin lesions task, all dataset targets reside at the leaf layer. As biological taxonomies were not provided natively in iWildCam 2.0 or the Plankton datasets, we programmatically mapped class labels to their standardized hierarchical lineages using the GBIF Backbone Taxonomy API [11] and the WoRMS API [55], respectively. In all hold-out cases (skin lesions, fungi, plankton), leaf-stratified sampling is 10% of each leaf class, ensuring that minority classes remain represented in the validation set. Validation is used for early stopping on leaf-level balanced accuracy and for fitting the post-hoc temperature-scaling scalar (Section F.8).

To isolate the architectural contributions of our method, we evaluate the Parked Product Space against the following baselines:

Learnable Euclidean Baselines (128D & 512D). Standard Euclidean networks optimized via cross-entropy over scaled cosine similarities between L_2-normalized embeddings and learned class prototypes [51]. By dynamically learning their prototypes across embedding dimensions (128D and 512D), these baselines are granted maximum capacity to interpolate fine-grained classes [30]. *Metric-Guided Prototypes (MGP).* A hierarchy-aware Euclidean baseline [10] with learned prototypes in a 512D embedding, logits given by negative squared Euclidean distances, and a scale-free distortion regularizer that aligns pairwise prototype distances with the taxonomic target distances that define our h-MDS prototypes (Section E), giving both methods an identical hierarchical prior; the regularization weight

[1] For the camera-ready version, the plankton data were re-obtained after a change in the public download API; all plankton results were recomputed for every model (Section F.1).

Table 2: Balanced accuracy (%) at L1 and leaf level on four OOD benchmarks, mean ± std over 5 random seeds. OOPark uses N sub-manifolds of 20D ($N\times$20D: 40D ISIC, 60D iWildCam, 100D Fungi, 80D Plankton); S. Lorentz $20N$ is a single manifold of matched total dimensionality. The unparked product ablation is shown in Fig. 2a; intermediate levels are reported in Section B.

	ISIC		**iWildCam**		**Fungi**		**Plankton**	
Model	L1	Leaf	L1	Leaf	L1	Leaf	L1	Leaf
OOPark $N \times 20$	**68.5**±1.3	**59.7**±1.4	**59.2**±1.8	**45.9**±1.4	**91.5**±1.1	58.2±0.8	**64.6**±2.8	**41.3**±3.3
S. Lorentz 20	63.1±1.4	54.6±2.5	56.8±2.3	43.8±2.2	86.4±0.9	51.6±1.5	63.1±4.1	30.2±5.0
S. Lorentz $20N$	62.4±1.4	55.1±1.8	55.6±2.1	45.7±1.5	84.7±1.2	53.2±1.2	59.3±1.2	38.3±2.1
Euclidean 512	66.2±1.8	55.2±2.6	56.6±2.3	44.8±1.2	89.1±1.0	**61.5**±0.8	61.7±2.3	37.8±3.1
Euclidean 128	65.3±1.4	54.6±2.1	56.9±4.2	38.5±2.8	87.4±1.3	59.3±1.1	59.8±2.4	32.5±3.6
MGP 512	66.8±1.9	54.9±2.6	56.6±3.9	43.6±2.6	85.0±0.6	58.3±0.5	63.0±5.0	35.2±3.2

is selected on validation ($\beta = 0.1$). *Single Lorentz Space.* A single-manifold hyperbolic baseline ($\mathbb{L}_k^d$) using the exact same fixed, pre-computed h-MDS prototypes as our product space, but forcing the entire taxonomy into a single, undivided volume. The overarching taxonomic root is anchored at the mathematical origin. This baseline isolates the need for decoupling the representation space to resolve angular crowding. Because high-dimensional Lorentz optimization is prone to numerical instability and gradient overflow [20, 33], we use 20 dimensions per Lorentz manifold and compare against single Lorentz manifolds of both 20D (S. Lorentz 20) and OOPark's total dimensionality (S. Lorentz $20N$, 40–100D); the latter separates the effect of the product structure from that of capacity.

Implementation Details Hierarchical prototypes are pre-computed offline using Hyperbolic Multi-dimensional Scaling (h-MDS) [42], optimized to preserve both taxonomic hop-distance and frequency-based volumetric margins, then permanently locked (Section E). Visual features are extracted using a ConvNeXt-Tiny backbone [29] for all main results, with EfficientNetV2 [47], Swin [28] and ResNet [16] used in the cross-architecture study (Section F.3), initialized with standard pre-trained weights via `timm` [53], followed by an MLP projecting latents into the target dimensionality. Parameter counts and training and inference costs for all models are reported in Section F.4. Hyperbolic operations are implemented using `geoopt` [21]. All network parameters, including manifold components, are optimized end-to-end using standard AdamW; Euclidean updates are projected onto the Lorentz manifold via the exponential map (`expmap`) [5, 20]. Newly initialized projections use a higher learning rate than the backbone [18], and curvature parameters are initialized at $k = 1$. All models use early stopping based on peak leaf-level validation performance. Hyperparameters, augmentations, and hardware specifications are deferred to Section F. Long-tailed class imbalance is handled identically for all models by combining inverse-frequency sampling with fractional loss weighting (Section F.2). Confidence calibration is assessed via post-hoc temperature scaling [14]: a single scalar temperature parameter is optimized on the pre-softmax validation logits via linear search.

5 Results

All results are reported as mean and standard deviation over five random seeds, unless otherwise specified. Across four distribution shifts (Table 2), OOPark achieves the highest L1 balanced accuracy on every benchmark (by 1.5–2.4 pp over the strongest baseline) and the highest leaf accuracy on ISIC and Plankton (by 4.5 and 3.0 pp). At the leaf, it is level with the matched single manifold on iWildCam and trails the Euclidean baselines on Fungi (Section 6). Enlarging the single manifold from 20 to $20N$ dimensions raises its leaf accuracy on every benchmark but lowers its L1 accuracy, so it remains behind OOPark at L1 throughout. Stratifying test samples by embedding radius locates OOPark's advantage over the matched single manifold near the origin, where angular crowding acts (Fig. 2b): on iWildCam its leaf accuracy is 2.2× higher in the lowest-radius decile, falling to 1.07× over all samples (Plankton: 1.7× and 1.08×; per-dataset curves in Section A).

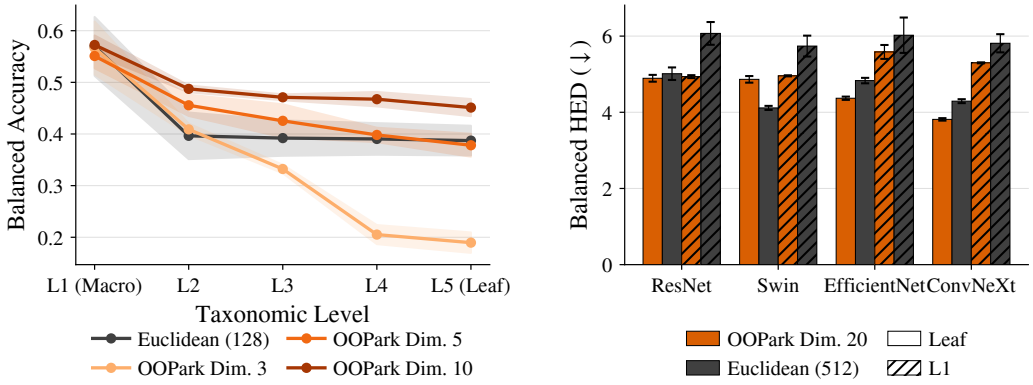
To assess whether structural preservation generalizes across architectures, we evaluate the Hierarchical Error Distance (HED) of misclassifications, the lowest-common-ancestor hop-distance between the predicted and ground-truth classes [3], across diverse feature extractors (Fig. 1b). We report a class-balanced variant, bHED, macro-averaged over leaf classes (leaf metric) or L1 macro-classes (root-level metric), so long-tailed leaf frequencies do not dominate. The 512D Euclidean baseline

exhibits substantial L1 error gaps across convolutional networks (ResNet, EfficientNet, ConvNeXt). The Swin Vision Transformer shows a different trade-off, plausibly because self-attention lacks the localized hierarchical filters of CNNs [40]. OOPark ($d = 20$) consistently prevents L1 macro-collapse across all encoders, indicating it functions as a robust structural regularizer.

Parametric Efficiency in Orthogonal Sub-spaces To isolate the contribution of orthogonal decoupling from raw dimensional capacity, we sweep per-sub-manifold dimensionality $d_m \in \{3, 5, 10\}$ on iWildCam (Fig. 1a). Two patterns emerge. First, macro-level (L1) balanced accuracy is essentially flat across d_m, indicating that the orthogonal product structure, not sub-manifold dimensionality governs macro-routing. Second, fine-grained accuracy at deeper ranks (L2–L5) scales with d_m, as expected for species-level discrimination that consumes intra-manifold volume. A product of three 10-dimensional Lorentz sub-manifolds (3×10 active dimensions) matches or exceeds the 128D Euclidean baseline at every taxonomic level, despite using less than a quarter of the embedding dimensionality (Fig. 1a). This confirms the volume-growth argument of Section 3: a low-dimensional Lorentz sub-manifold supplies sufficient hyperbolic volume for species-level discrimination, whereas the Euclidean baseline must spend dimensions to compensate for polynomial volume growth [26].

Confidence Calibration and Structural Reliability We assess calibration and misclassification detection under distribution shift on the iWildCam OOD split (Table 3). We report Expected Calibration Error (ECE) and AUROC for misclassification detection using Maximum Softmax Probability (MSP) and the intrinsic geometric radius, under both Micro-averaged (sample-pooled) and L1 Macro-averaged (class-balanced) settings. While the 512D Euclidean baseline achieves the lowest Micro-ECE, this is largely an artifact of the dominant *Mammalia* clade. Under L1 Macro-averaging, its ECE rises to 0.23, indicating overconfidence on minority taxonomic branches. OOPark achieves the highest L1 Macro-AUROC for MSP and, crucially, the strongest geometric radius signal for error detection, outperforming both the Euclidean MSP baseline and the single-manifold models (Table 3).

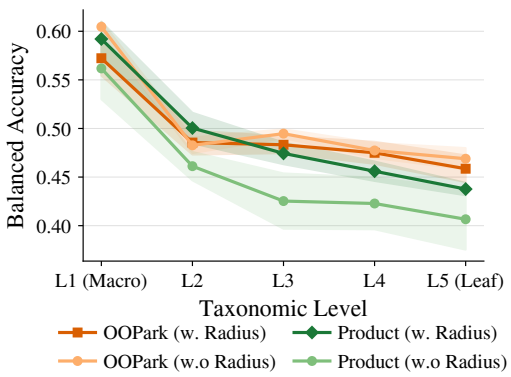
The Necessity of Origin Parking To isolate the architectural necessity of the parking mechanism, we compare OOPark against the unparked product space, each with and without the radius penalty (Figure 2a). Without parking, the product space drifts representationally, degrading fine-grained accuracy. A global radius penalty on the unparked space improves full taxonomic performance, while having an adverse effect on OOPark, which asymmetrically regularizes inactive branches, preserving capacity and performance across the entire hierarchy. Tracking tangent-space gradients during training, single-manifold models show gradient spikes and high variance, the unparked product



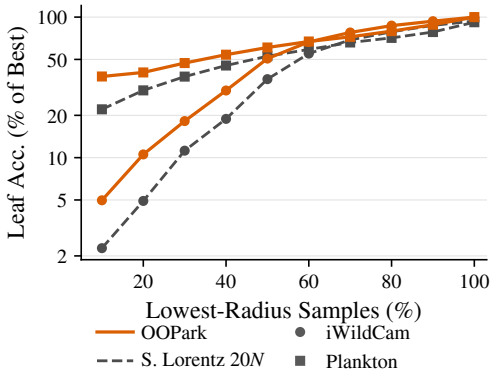
(a) Balanced accuracy across taxonomic levels for varying sub-manifold dimensions. Shaded bands denote ± 1 std. across seeds.

(b) Balanced Hierarchical Error Distance (bHED) across visual encoders at macro (L1) and leaf levels. Error bars denote $\pm$ 1 std across seeds.

Figure 1: **Dimensionality ablation and cross-architecture structural fidelity** on the iWildCam dataset. (a) Performance impact of scaling the active sub-manifold dimensionality against a Euclidean baseline. (b) Comparison of global (L1) and local (Leaf) structural preservation (bHED, lower is better) between OOPark and a Euclidean baseline across varied CNN and Transformer backbones.



(a) Balanced accuracy across taxonomic levels on iWildCam. Shaded bands denote ± 1 std. over seeds.



(b) Cumulative leaf accuracy over radii, relative to the best overall accuracy per dataset (log scale).

Figure 2: **Origin parking and angular crowding.** (a) Performance of OOPark and the unparked variation with (w.) and without (w.o) radius penalty (Radius). (b) OOPark versus a single Lorentz manifold of matched dimensionality; the gap is largest near the origin, where angular crowding acts.

space shows vanishing gradients, and OOPark maintains a stable mean gradient with low variance (Section C).

6 Discussion

Macro-Decoupling Versus Leaf-Level Capacity Across our four benchmarks, OOPark improves macro-level (L1) balanced accuracy over every baseline, including the hierarchy-aware MGP and a single manifold of matched dimensionality (Table 2). The leaf-level pattern is not uniform: OOPark wins on ISIC, iWildCam, and Plankton but trails both Euclidean baselines on Fungi and ties MGP there. The matched-capacity comparison separates the two levels: the single manifold of dimension $20N$ remains behind OOPark at L1 on every benchmark, but comes closest at the leaf, where it matches OOPark on iWildCam. Macro-level robustness therefore stems from the decoupling rather than from capacity, whereas leaf-level accuracy is partly capacity-driven. Macro-decoupling provides a structural prior whose benefit appears robust to architectural choice and shift severity: routing L1 branches into independent sub-manifolds neutralizes this failure mode of single-manifold hyperbolic geometries, in which angular crowding compounds with representational drift to push fine-grained classes across decision boundaries. Leaf-level discrimination, by contrast, depends on whether the shift pushes features out of their training-time neighborhoods. The leaf margin also narrows as the number of leaf classes grows, but this trend is confounded with shift severity: Fungi has by far the

Table 3: Comparison of Pooled (Standard) vs. L1 Macro-averaged metrics on iWildCam. Pooled metrics reflect the total test set performance, whereas L1 Macro metrics provide a class-balanced view across the primary taxonomic branches. Product denotes the unparked product space, Singl. Lorentz the single manifold setting, and Euclid. the Euclidean prototype models. **MSP** (Maximum Softmax Probability) and **Radius** (hyperbolic radius) denote the AUROC for misclassification detection using these metrics as confidence scores. Radius is not defined for the Euclidean models; — used.

	Micro-Averaged			L1 Macro-Averaged		
Model	**ECE** ↓	**MSP** ↑	**Radius** ↑	**ECE** ↓	**MSP** ↑	**Radius** ↑
Euclid. (512)	**0.06** $\pm$ 0.01	0.91 $\pm$ 0.01	—	0.23 $\pm$ 0.02	0.74 $\pm$ 0.06	—
Euclid. (128)	0.06 $\pm$ 0.01	0.85 $\pm$ 0.03	—	0.22 $\pm$ 0.09	0.74 $\pm$ 0.12	—
MGP (512)	0.09 $\pm$ 0.03	0.81 $\pm$ 0.05	—	0.17 $\pm$ 0.04	0.75 $\pm$ 0.06	—
S. Lorentz (20)	0.09 $\pm$ 0.00	0.95 $\pm$ 0.02	**0.95** $\pm$ 0.01	0.16 $\pm$ 0.00	0.88 $\pm$ 0.02	0.87 $\pm$ 0.02
S. Lorentz (60)	0.12 $\pm$ 0.01	**0.95** $\pm$ 0.03	0.94 $\pm$ 0.02	**0.14** $\pm$ 0.03	0.87 $\pm$ 0.03	0.77 $\pm$ 0.03
Product	0.14 $\pm$ 0.02	0.88 $\pm$ 0.05	0.88 $\pm$ 0.03	0.18 $\pm$ 0.04	0.85 $\pm$ 0.04	0.88 $\pm$ 0.03
OOPark	0.08 $\pm$ 0.01	0.94 $\pm$ 0.01	0.93 $\pm$ 0.02	0.17 $\pm$ 0.03	**0.89** $\pm$ 0.01	**0.92** $\pm$ 0.01

most leaf classes and exhibits the smallest validation-to-test drop in our study (Table 12; ΔL1 = 0.6 pp, indistinguishable val and test at L1), reflecting that the geographic Danish/non-Danish split does not induce strong visual shift. The high-capacity Euclidean baseline can therefore safely rely on feature interpolation to separate fine-grained classes without incurring the representational drift that OOPark is designed to mitigate; the structural overhead is therefore justified specifically when the distribution shift is severe.

Preserving the Radius-Certainty Link in a Product Space Single-manifold hyperbolic classifiers are also valued for an intrinsic property of the geometry: radial distance from the origin correlates with predictive certainty. The unparked variant confirms this property can be lost in a product space: on iWildCam, both Micro- and L1-Macro-Averaged Radius AUROCs collapse to 0.88, against 0.95/0.87 for the 20D single Lorentz baseline (Table 3), because inactive sub-manifold drift injects noise into the aggregate radius. OOPark restores the link (0.93/0.92 on iWildCam) and gives the strongest L1 Macro-Averaged Radius signal on iWildCam and Plankton, within 0.01 of the best on Fungi (Table 13, Table 14). The radius-stratified accuracies (Fig. 2b) complement this: against the matched single manifold, the radius ranks errors about equally well (Micro-Averaged AUROC 0.93 vs 0.94 on iWildCam, 0.71 vs 0.72 on Plankton), but OOPark makes fewer errors among the uncertain samples, with the entire accuracy gap lying in the lower half of the radius range. We do not propose OOPark as an uncertainty-estimation method; rather, parking preserves a geometric property that the unconstrained product space sacrifices.

Limitations and future work The current formulation of Orthogonal Origin Parking relies on a predefined, static macro-taxonomy and pre-computed hyperbolic prototypes; downstream performance therefore depends on the h-MDS stopping criterion, which trades macro- against leaf-level fidelity (Section E). The architecture's sensitivity to missing or erroneous annotations remains uncharacterized, although leaf accuracy on ISIC is insensitive to the choice of macro-split (Section D), and the fixed prototypes prevent the model from dynamically adjusting class centers to accommodate latent visual variance. Large sub-trees can be accommodated by increasing the sub-manifold dimension, since leaf accuracy scales with d_m (Fig. 1a). A more substantive scaling concern lies in the optimization dynamics rather than parameter count, which adds only a backbone-dependent few percent: the ground-truth term $\tilde{D}_y$ accumulates parking distances across all $N-1$ inactive sub-manifolds, and as N grows this parking sum can come to dominate the loss distance numerically. Hence, the loss landscape may be governed almost entirely by the radii in inactive sub-manifolds rather than by distance to the correct active prototype, attenuating the gradient signal that drives fine-grained intra-branch discrimination. Topologies with very broad root structures may require stochastic subsampling of the inactive parking terms to keep the active classification signal informative as N scales. Future work should explore dynamic end-to-end prototype learning, principled per-branch loss balancing for large N, decoupling below L1 for trees whose sub-branches differ in curvature (the learned curvatures already differ across macro-branches; Fig. 9), combining OOPark with domain-generalization objectives, evaluation on non-biological and non-visual hierarchies, and the precise alignment of orthogonal spaces with transformer-specific inductive biases.

7 Conclusion

With our Orthogonal Origin Parking, we show that partitioning macro-branches into isolated hyperbolic spaces mitigates the angular crowding of single-manifold geometries. By anchoring inactive components to the origin, the framework regulates gradient behavior during optimization and preserves the radius-certainty correlation typical of single-manifold hyperbolic models within a product space. Evaluated across four out-of-distribution benchmarks, the parked product space improves balanced accuracy at the macro level over the Euclidean baselines, the hierarchy-aware MGP, and a single manifold of matched dimensionality on all settings, while using 40–100 embedding dimensions against up to 512 for the Euclidean baselines; since the matched single manifold does not close this gap, the macro-level gains stem from the decoupling rather than from capacity. By maintaining global taxonomic fidelity when fine-grained visual features are degraded, this routing mechanism provides a structural pathway for classification models to preserve valid ancestral predictions in open-world settings.

References

[1] Gregor Bachmann, Gary Becigneul, and Octavian Ganea. "Constant Curvature Graph Convolutional Networks". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 486–496. URL: https://proceedings.mlr.press/v119/bachmann20a.html.

[2] Sara Beery, Elijah Cole, and Arvi Gjoka. "The iWildCam 2020 Competition Dataset". In: *CoRR* abs/2004.10340 (2020). arXiv: 2004.10340. URL: https://arxiv.org/abs/2004.10340.

[3] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. "Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12506–12515.

[4] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[5] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. "Hyperbolic Graph Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/0415740eaa4d9decbc8da001d3fd805f-Paper.pdf.

[6] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. "Fully Hyperbolic Neural Networks". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 5672–5686. DOI: 10.18653/v1/2022.acl-long.389. URL: https://aclanthology.org/2022.acl-long.389/.

[7] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C. Halpern, Susana Puig, and Josep Malvehy. "BCN20000: Dermoscopic Lesions in the Wild". In: (2019). arXiv: 1908.02288 [eess.IV]. URL: https://arxiv.org/abs/1908.02288.

[8] Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. "Large Margin Deep Networks for Classification". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018.

[9] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. "Hyperbolic Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/dbab2adc8f9d078009ee3fa810bea142-Paper.pdf.

[10] Vivien Sainte Fare Garnot and Loic Landrieu. "Leveraging Class Hierarchies with Metric-Guided Prototype Learning". In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2021.

[11] GBIF Secretariat. *GBIF Backbone Taxonomy*. Global Biodiversity Information Facility, 2023. DOI: 10.15468/39omei. URL: https://doi.org/10.15468/39omei.

[12] Mina Ghadimi Atigh, Martin Keller-Ressel, and Pascal Mettes. "Hyperbolic Busemann Learning with Ideal Prototypes". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 103–115. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/01259a0cb2431834302abe2df60a1327-Paper.pdf.

[13] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Re. "Learning mixed-curvature representations in product manifolds". In: *International Conference on Learning Representations (ICLR)*. 2019.

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. "On Calibration of Modern Neural Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 1321–1330. URL: https://proceedings.mlr.press/v70/guo17a.html.

[15] Zihao Guo, Qingyun Sun, Haonan Yuan, Xingcheng Fu, Min Zhou, Yisen Gao, and Jianxin Li. "GraphMoRE: Mitigating Topological Heterogeneity via Mixture of Riemannian Experts". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 39.11 (Apr. 2025), pp. 11754–11762. DOI: 10.1609/aaai.v39i11.33279. URL: https://ojs.aaai.org/index.php/AAAI/article/view/33279.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[17] Dan Hendrycks and Thomas Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *International Conference on Learning Representations (ICLR)*. 2019.

[18] Jeremy Howard and Sebastian Ruder. "Universal Language Model Fine-tuning for Text Classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031. URL: https://aclanthology.org/P18-1031/.

[19] Laetitia Jalabert, Guillaume Signoret, Louis Caray-Counil, Marion Vilain, Emmanuelle Martins, Fabien Lombard, Marc Picheral, and Jean-Olivier Irisson. *FlowCAMNet: plankton images captured with the FlowCAM*. 2024. DOI: 10.17882/101961. URL: https://doi.org/10.17882/101961.

[20] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. "Hyperbolic Image Embeddings". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[21] Max Kochurov, Rasul Karimov, and Serge Kozlukov. *Geoopt: Riemannian Optimization in PyTorch*. 2020. arXiv: 2005.02819 [cs.CG].

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. "WILDS: A Benchmark of in-the-Wild Distribution Shifts". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 5637–5664. URL: https://proceedings.mlr.press/v139/koh21a.html.

[23] J. B. Kruskal. "Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis". In: *Psychometrika* 29.1 (1964), pp. 1–27. DOI: 10.1007/BF02289565.

[24] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. "Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution". In: *International Conference on Learning Representations (ICLR)*. 2022.

[25] Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. "Lorentzian Distance Learning for Hyperbolic Representations". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 3672–3681. URL: https://proceedings.mlr.press/v97/law19a.html.

[26] Nathan Linial, Eran London, and Yuri Rabinovich. "The geometry of graphs and some of its algorithmic applications". In: *Combinatorica* 15.2 (1995), pp. 215–245.

[27] Qi Liu, Maximillian Nickel, and Douwe Kiela. "Hyperbolic Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10012–10022.

[29] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.

[30] Pascal Mettes, Elise van der Pol, and Cees Snoek. "Hyperspherical Prototype Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/02a32ad2669e6fe298e607fe7cc0e1a0-Paper.pdf.

[31] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. "How to Exploit Hyperspherical Embeddings for Out-of-Distribution Detection?" In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=aEFaE0W5pAd.

[32] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. "No Fuss Distance Metric Learning using Proxies". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 360–368.

[33] Maximillian Nickel and Douwe Kiela. "Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 3779–3788. URL: https://proceedings.mlr.press/v80/nickel18a.html.

[34] Maximillian Nickel and Douwe Kiela. "Poincaré Embeddings for Learning Hierarchical Representations". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf.

[35] Eric C. Orenstein, Oscar Beijbom, Emily E. Peacock, and Heidi M. Sosik. "WHOI-Plankton- A Large Scale Fine Grained Visual Recognition Benchmark Dataset for Plankton Classification". In: (2015). arXiv: 1510.00745 [cs.CV]. URL: https://arxiv.org/abs/1510.00745.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[37] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. "Hyperbolic Deep Neural Networks: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.12 (2022), pp. 10023–10044. DOI: 10.1109/TPAMI.2021.3136921.

[38] Lukas Picek, Klara Janouskova, Vojtech Cermak, and Jiri Matas. "FungiTastic: A Multi-Modal Dataset and Benchmark for Image Categorization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2025, pp. 2071–2081.

[39] Lukáš Picek, Milan Šulc, Jiří Matas, Thomas S. Jeppesen, Jacob Heilmann-Clausen, Thomas Læssøe, and Tobias Frøslev. "Danish Fungi 2020 - Not Just Another Image Recognition Dataset". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2022, pp. 1525–1535.

[40] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. "Do Vision Transformers See Like Convolutional Neural Networks?" In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 12116–12128. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/652cf38361a209088302ba2b8b7f51e0-Paper.pdf.

[41] John G Ratcliffe. *Foundations of Hyperbolic Manifolds*. Vol. 149. Springer Science & Business Media, 2006.

[42] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. "Representation Tradeoffs for Hyperbolic Embeddings". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 4460–4469. URL: https://proceedings.mlr.press/v80/sala18a.html.

[43] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. "How Does Batch Normalization Help Optimization?" In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/905056c1ac1dad141560467e0a99e1cf-Paper.pdf.

[44] Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. "Mixed-curvature Variational Autoencoders". In: *International Conference on Learning Representations (ICLR)*. 2020.

[45] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical Networks for Few-shot Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

[46] Li Sun, Zhongbao Zhang, Junda Ye, Hao Peng, Jiawei Zhang, Sen Su, and Philip S Yu. "A Self-Supervised Mixed-Curvature Graph Neural Network". In: vol. 36. 4. June 2022, pp. 4146–4155. DOI: 10.1609/aaai.v36i4.20333. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20333.

[47] Mingxing Tan and Quoc Le. "Efficientnetv2: Smaller models and faster training". In: *International conference on machine learning*. PMLR. 2021, pp. 10096–10106.

[48] Niels K. Ternov, Anders N. Christensen, Peter J. T. Kampen, Gustav Als, Tine Vestergaard, Lars Konge, Martin Tolsgaard, Lisbet R. Hölmich, Pascale Guitera, Annette H. Chakera, and Morten R. Hannemose. "Generalizability and usefulness of artificial intelligence for skin cancer diagnostics: An algorithm validation study". In: *JEADV Clinical Practice* 1.4 (2022), pp. 344–354. DOI: https://doi.org/10.1002/jvc2.59. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jvc2.59. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jvc2.59.

[49] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific Data* 5.1 (2018), pp. 1–9.

[50] Jack Valmadre. "Hierarchical classification at multiple operating points". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022, pp. 25528–25541.

[51] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. "NormFace: L2 Hypersphere Embedding for Face Verification". In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM '17. Mountain View, California, USA: Association for Computing Machinery, 2017, pp. 1041–1049. ISBN: 9781450349062. DOI: 10.1145/3123266.3123359. URL: https://doi.org/10.1145/3123266.3123359.

[52] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. "A Discriminative Feature Learning Approach for Deep Face Recognition". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 499–515. ISBN: 978-3-319-46478-7.

[53] Ross Wightman. *PyTorch Image Models*. https://github.com/rwightman/pytorch-image-models. 2019.

[54] Ross Wightman, Hugo Touvron, and Hervé Jégou. *ResNet strikes back: An improved training procedure in timm*. 2021. arXiv: 2110.00476 [cs.CV].

[55] WoRMS Editorial Board. *World Register of Marine Species*. 2024. DOI: 10.14284/170. URL: https://www.marinespecies.org.

A Angular crowding

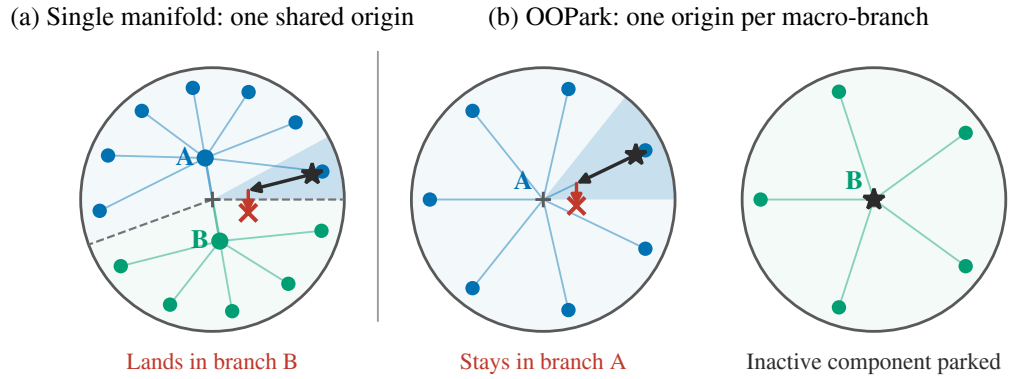


Figure 3: Schematic of angular crowding, drawn in the Poincaré disk for visualisation. (a) In a single manifold, both macro-branches share the angular budget around one origin. A sample of a branch-A class (star) contracts toward the origin under distribution shift (black arrow), where its class sector (shaded) spans only a short arc, so a small perturbation (red) carries it into branch B. (b) OOPark gives each macro-branch its own sub-manifold, with the L1 node at the origin and the full angular budget available to its classes. The sample's component in sub-manifold B is parked at the origin, and the same perturbation only reaches a sibling class within branch A.

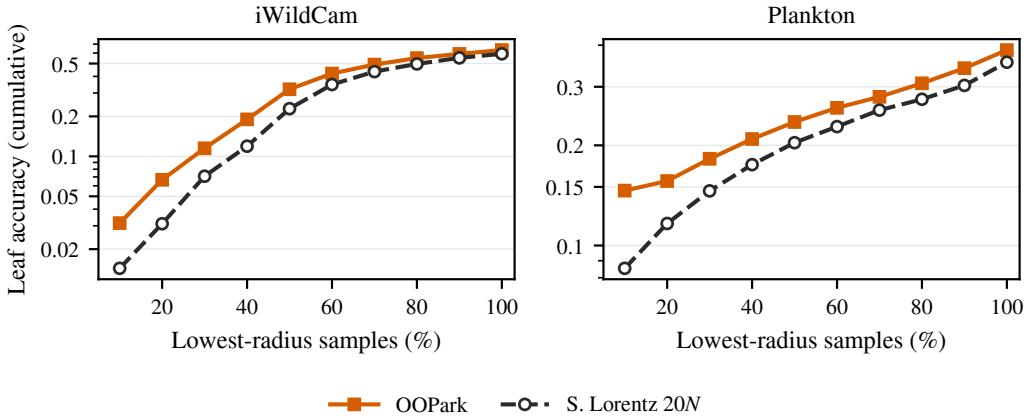


Figure 4: Cumulative leaf accuracy over the lowest-radius X% of test samples on iWildCam and Plankton (single run each), for OOPark and a single Lorentz manifold of matched dimensionality (S. Lorentz $20N$: 60D and 80D). Using percentiles makes the comparison invariant to the models' radius scales. Accuracies are raw rather than balanced, so the 100% values differ from Table 2. On both benchmarks, OOPark's advantage is largest near the origin and shrinks toward the periphery (iWildCam: 2.2× in the lowest-radius decile versus 1.07× over all samples; Plankton: 1.7× versus 1.08×), as angular crowding predicts.

B Per-level results

Figure 5 reports balanced accuracy at every taxonomic level, complementing the L1 and leaf columns of Table 2, with which its endpoints coincide. OOPark equals or exceeds the 512D Euclidean baseline at every level on iWildCam and Plankton, and at L1, where it concedes the finer levels to MGP, consistent with the mild shift of that benchmark (Table 12). The skin-lesion OOD set has depth 2, so Table 2 already contains its full breakdown; it is included here for completeness.

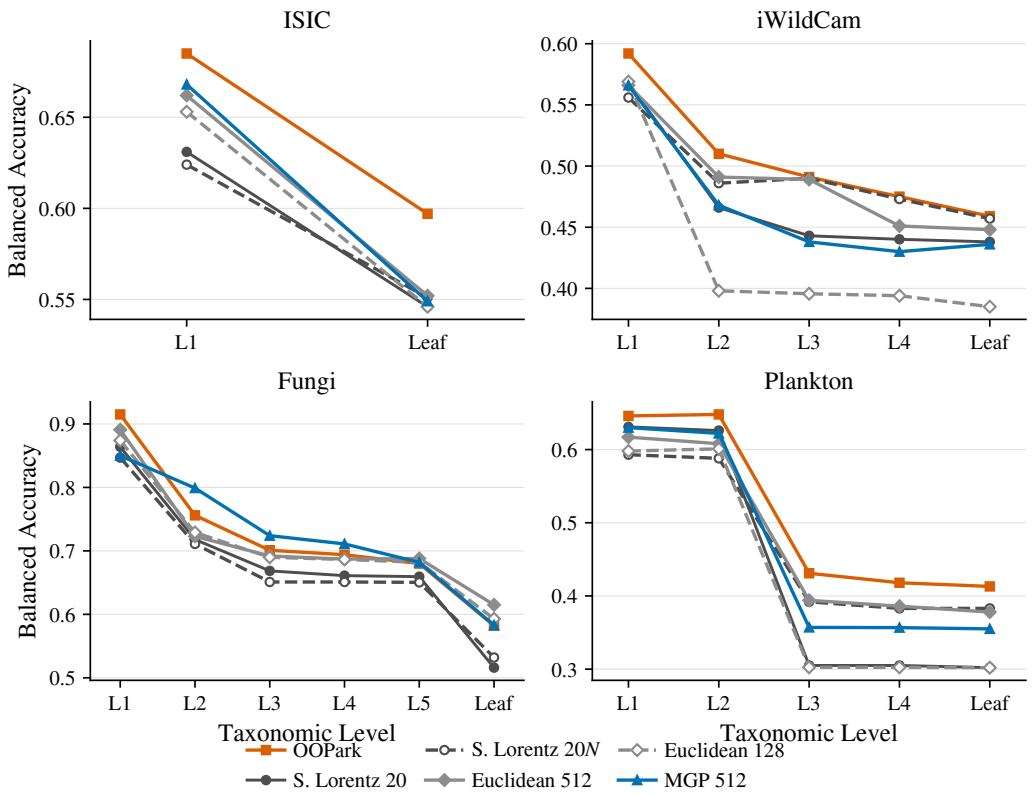


Figure 5: Balanced accuracy at every taxonomic level on the four OOD benchmarks, mean over 5 seeds. Shaded bands denote ± 1 std. where it is available at every level The L1 and leaf values coincide with Table 2.

C Gradient dynamics

We track the mean and variance of tangent feature spatial gradients ($\nabla_{\mathbf{v}}\mathcal{L}$) during training (Figure 6). Single-manifold models exhibit gradient spikes and high variance, a well known pathology of Lorentz optimization [20, 33]. The unconstrained product space suffers from vanishing gradients: inactive branches drift into empty hyperbolic space, removing the spatial tension needed for a meaningful gradient signal. OOPark anchors inactive components to the origin, establishing a geometric equilibrium. Consistent with findings on optimization landscape smoothness [43], the asymmetric distance maintains a healthy mean gradient signal while suppressing variance, avoiding both the volatility of a single manifold and the near-zero gradients of the unparked product space.

D Choice of macro-split on skin lesions

The default L1 split for skin lesions is benign versus malignant. Replacing it with a melanocytic versus non-melanocytic split, with prototypes recomputed by h-MDS and all other settings fixed, gives a leaf balanced accuracy of 59.1 ± 1.6, against 59.7 ± 1.4 for the default split. L1 accuracy under the alternative split (74.0 ± 2.1) measures a different macro task and is not comparable.

E Hyperbolic Prototype Generation (h-MDS)

Because the Orthogonal Origin Parking architecture relies on fixed geometric anchors to enforce the structural penalty (Section 3), the hierarchical prototypes must be pre-computed prior to training the visual encoder. To embed the discrete biological taxonomy into the continuous Lorentz manifolds, we utilize Hyperbolic Multi-Dimensional Scaling (h-MDS) [42].

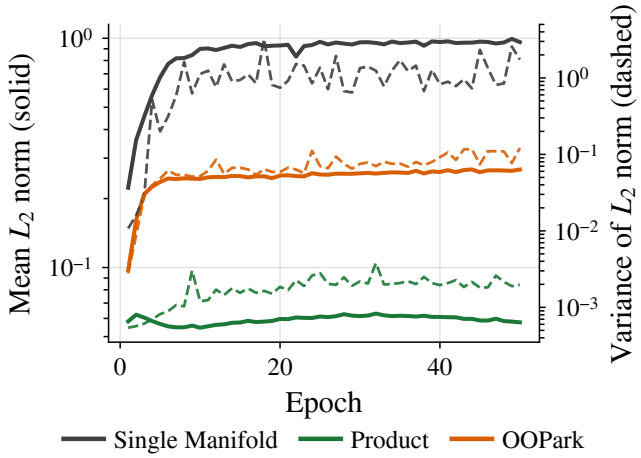


Figure 6: Seed-averaged mean and variance of the gradient norm $\|\nabla_{\mathbf{v}}\mathcal{L}\|_2$ during training on iWildCam (log scale, twin axes) for the three hyperbolic variants.

Given the dataset taxonomy, we first define an ideal pairwise distance matrix D^* for all nodes. The target distance between any two taxonomic nodes i and j is formulated as a combination of their lowest-common-ancestor (LCA) hop distance and a frequency-based volumetric margin:

$$D^*_{i,j} = \Big(\text{Hops}(i,j) \times \lambda\Big) + \gamma \left(\frac{\log(C_i+1) + \log(C_j+1)}{\log(C_{\max}+1)}\right) \tag{11}$$

where λ is a hyperparameter scaling the baseline branch length, γ is the margin weight, and C_i is the empirical training frequency of class i. Crucially, this logarithmic margin ensures that massive majority classes are allocated larger target distances, effectively preventing structural overcrowding.

To map these target distances to the spatial coordinates of the m-th Lorentz sub-manifold ($\mathbb{L}^{d_m}_{k_m}$), we initialize the node embeddings near the origin. To prevent spatial drift and strictly align the taxonomy with the parking mechanism, the root node of the sub-tree is pinned to the mathematical origin of the manifold (i.e., the base of the hyperboloid). The coordinates of the remaining descendant nodes are then optimized via Riemannian gradient descent to minimize the Mean Squared Error (MSE) between the ideal graph distances and the computed hyperbolic distances:

$$\mathcal{L}_{\text{MDS}} = \frac{1}{|V|^2} \sum_{i \in V} \sum_{j \in V} \left(\sqrt{k_m}\, \text{arcosh}\left(-\frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle_{\mathcal{L}}}{k_m}\right) - D^*_{i,j}\right)^2 \tag{12}$$

Once this optimization converges, the resulting spatial coordinates are extracted and frozen as the fixed target prototypes $\mathbf{c}_i$ defined in Equation 7 [20].

Sensitivity to prototype placement To test whether performance depends on the exact placement of the prototypes, we scale the branch-length parameter λ by $\{0.75, 1, 1.25\}$ and vary the frequency-margin weight $\gamma \in \{0.3, 0.5, 0.8\}$ around the defaults used throughout, recomputing the h-MDS prototypes for each of the nine configurations (Fig. 7). Due to compute constraints, each configuration is trained on Plankton for 15 of the 100 epochs with the unchanged cosine schedule (one seed) and evaluated on the OOD test split after every epoch; the defaults were fixed beforehand and no configuration is selected from this sweep. L1 accuracy reaches its final level within the first epochs and fluctuates around the reported value for every configuration. Leaf accuracy rises toward the reported value in all configurations, reaching 33.6–38.2% (mean over epochs 11–15) against 41.3% for the reported 100-epoch model. The spread across configurations (5.4 pp at L1, 5.6 pp at the leaf) is comparable to the epoch-to-epoch fluctuation within a single run) and to the seed standard deviation of the reported model (2.8 and 3.3 pp), so performance shows no sensitivity to placement within this range beyond run-to-run variation.

Optimization Dynamics and Hierarchical Trade-offs During the prototype generation process, we empirically observed that the stopping criterion for the Riemannian optimization significantly

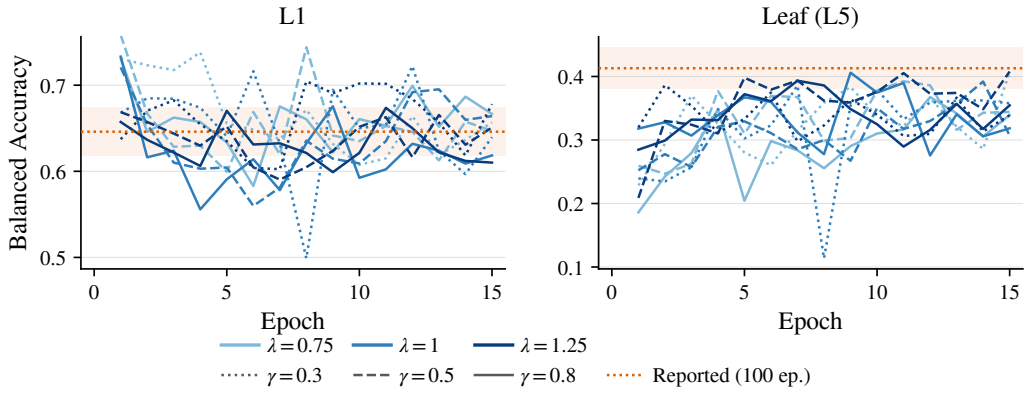


Figure 7: Sensitivity to the h-MDS placement hyperparameters on Plankton. Balanced accuracy on the OOD test split during the first 15 epochs for nine combinations of the branch-length scale λ (colour, as a multiple of the default) and the frequency-margin weight γ (line style), one seed each. The dotted line and shaded band show the reported 100-epoch result (mean ± 1 std over 5 seeds, Table 2).

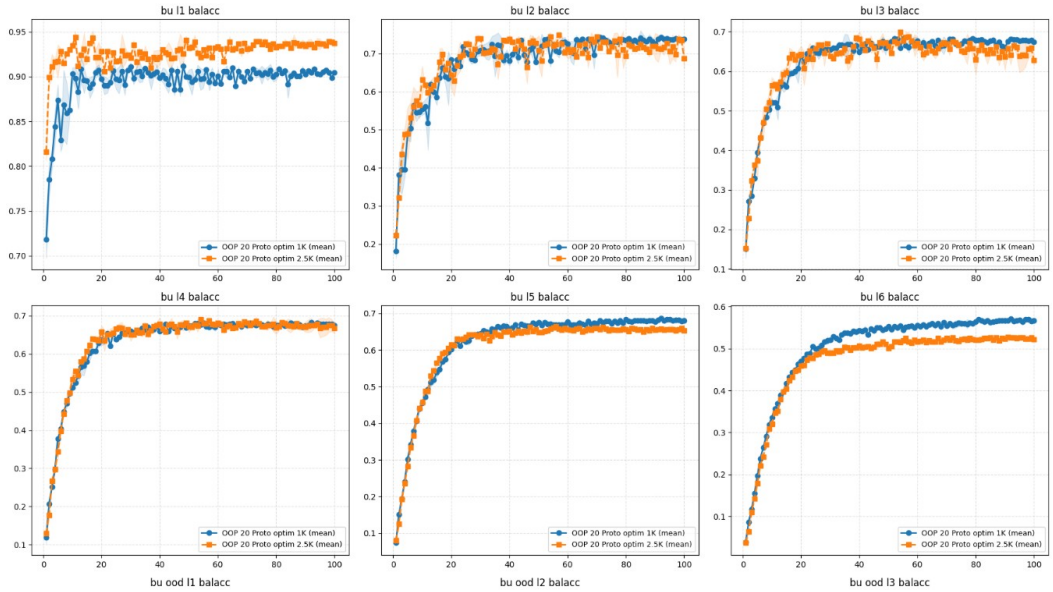


Figure 8: Fungi validation set performance for two choices of number of epochs in h-MDS training. (1K epochs and 2.5K epochs)

influences the downstream classification performance across different taxonomic ranks (L1–L6). The number of MDS epochs dictates the balance between broad macro-level separation and fine-grained leaf-level spacing. For all experiments presented in this work, the MDS optimization was systematically halted after 1000 epochs to establish a standardized structural baseline. However, extended optimization (e.g., to 2500 epochs) noticeably alters the distribution of representational capacity across the hierarchy. As illustrated in Figure 8, the optimal stopping strategy is therefore not absolute, but rather highly dependent on which specific taxonomic layer requires the highest predictive fidelity for the target application.

F Experimental configurations

To optimize the network, we utilized the AdamW optimizer with a differential learning rate and weight decay strategy tailored to specific parameter groups. The trainable parameters were partitioned

into three main components: the visual backbone, the classification projection heads, and the manifold curvature parameters.

To ensure stable optimization, weight decay was selectively applied. Specifically, weight decay was strictly limited to multi-dimensional weight tensors ($\geq$ 2D). One-dimensional tensors—including biases, normalization layers, and specific transformer embeddings (such as the class token and positional embeddings)—were entirely exempt from weight decay. Furthermore, weight decay was explicitly disabled for the trainable hyperbolic curvature parameters to prevent artificial flattening of the manifold geometry.

Each component was updated using specific learning rate multipliers: the randomly initialized projection heads were trained at the base learning rate, the manifold curvature parameters were optimized at a moderately reduced rate to ensure smooth geometric transitions, and the visual backbone (when unfrozen) was fine-tuned at a heavily reduced rate to preserve its pre-trained representational capacity.

Table 4: Optimization hyperparameters used for training.

Hyperparameter	**Value**
Optimizer	AdamW
Epochs	100
Batch size	16
Scheduler	Cosine Annealing ($T_{\max} = 101$)
Backbone Learning Rate	1×10^{-5}
Classification Head Learning Rate	1×10^{-3}
Curvature Learning Rate	1×10^{-4}
Classification Head Weight Decay	1×10^{-2}
Backbone Weight Decay	0.0
Curvature Weight Decay	0.0
Temperature init τ	1.0
Radius penalty α	0.05
target_depth	4
λ	target_depth / max_depth
γ	0.5

F.1 Validation splits

For skin lesions, validation uses a 10% hold-out of AISC, since complete taxonomic labels are available only on that source. For iWildCam, we use the canonical WILDS `ood_val` split [22]. For fungi, validation uses a 10% sample of the Danish-only subset of FungiTastic, keeping validation in the source geography so that the held-out test shift remains purely geographic. For plankton, validation uses a 10% hold-out of WHOI. All hold-outs are leaf-stratified at 10% of each leaf class, ensuring that minority classes remain represented in the validation set.

F.2 Long-Tailed Rebalancing Strategy

To address the long-tailed distribution of the training data without inducing severe overfitting on minority classes or unstable gradient spikes, we employ a decoupled fractional balancing strategy. We simultaneously apply a smoothed inverse-frequency sampling probability ($p \propto k^{-0.7}$) to the dataloader and a fractional inverse-frequency weight ($w \propto k^{-0.4}$) to the cross-entropy loss, where k is the class frequency. This effectively yields an expected gradient contribution that scales with $k^{-1.1}$, safely upweighting minority classes while distributing the regularization burden between sample repetition and gradient magnitude. This exact balancing configuration was held constant across all evaluated architectures and baselines to ensure a fair comparative analysis.

F.3 Architecture Initialization and Projection Heads

We detail the initialization of the Orthogonal Origin Parking architecture. All visual backbones (e.g., ConvNeXt, Swin) are initialized using pre-trained weights with the terminal classification heads removed. The base features are extracted at the final hidden dimension (D_{in}).

The used backbones are all loaded from torch-image-model (timm) with the strings:

Table 5: Summary of model architectures and their corresponding `timm` identifiers used in our experiments.

Model Architecture	`timm` **Identifier**
ResNet-50	`resnet50`
Swin Transformer (Tiny)	`swin_tiny_patch4_window7_224`
EfficientNetV2 (Small)	`tf_efficientnetv2_s.in21k_ft_in1k`
ConvNeXt (Tiny)	`convnext_tiny`

Manifold Setup The product space consists of K independent Lorentz sub-manifolds. Each sub-manifold is assigned a learnable curvature parameter k_i, uniformly initialized to $k = 1.0$. Because the d-dimensional Lorentz model requires one time-like coordinate, the visual features must be projected into a spatial dimension of $d - 1$.

Projection Heads To map the flat backbone features into the respective hyperbolic spaces, we construct K parallel, independent projection heads. To accommodate the different inductive biases of convolutional and attention-based networks, the normalization layer within the projection head is conditionally assigned: we utilize 1D Batch Normalization for CNN backbones and Layer Normalization for Transformer backbones. The exact layer sequence is detailed in Table 6.

Furthermore, to stabilize early optimization in Transformer-based backbones, the weights of the final linear projection layer are scaled by a factor of 0.01, and the biases are initialized strictly to zero.

Table 6: Architecture of the independent sub-manifold projection heads. D_{in} denotes the backbone's final hidden dimension, and d denotes the target embedding dimension of the Lorentz sub-manifold.

Layer	**Type / Configuration**	**Output Dimension**
Input	Extracted Backbone Features	D_{in}
1	Linear	$\min(512, D_{\text{in}})$
2	GELU Activation	$\min(512, D_{\text{in}})$
3	BatchNorm1d (CNNs) *or* LayerNorm (Transformers)	$\min(512, D_{\text{in}})$
4	Linear (Weight scaled by 0.01 for Transformers)	$d - 1$

Forward Pass and Tangent Space Stabilization To map the $d - 1$ dimensional Euclidean features generated by the projection heads onto the d-dimensional Lorentz sub-manifolds, we first construct tangent vectors at the mathematical origin. For a given spatial output $v_i \in \mathbb{R}^{d-1}$, we concatenate a zero to the time-like coordinate to form the tangent vector $\mathbf{v}_i = [0, v_i]$.

Because hyperbolic distance grows exponentially with respect to the tangent norm, unconstrained spatial projections can cause severe numerical instability (e.g., representation collapse or gradient explosion) when passed through the exponential map. To stabilize optimization, we apply a hard clipping mechanism to the tangent vectors prior to manifold projection. We constrain the L_2 norm of $\mathbf{v}_i$ by a predefined maximum radius $R = 15$:

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i \min\left(1, \frac{R}{\|\mathbf{v}_i\|_2 + \epsilon}\right)$$

where $\epsilon = 10^{-6}$ is a small constant to prevent zero-division. The stabilized tangent vector $\tilde{\mathbf{v}}_i$ is then safely projected onto the active Lorentz sub-manifold via the exponential map.

F.4 Compute Resources and Software Environment

All models, including baselines and the proposed Orthogonal Origin Parking architecture, were trained using NVIDIA L40s GPUs (16 workers). The computational environment was standardized across all experiments to ensure fair comparisons of training dynamics and convergence times. Requires <24GB VRAM.

The models and training pipelines were implemented in Python utilizing the PyTorch deep learning framework (v2.10.0) [36]. Visual backbones (ConvNeXt, ViT, Swin) were initialized using the `timm` library (v1.0.25) [54], and data augmentation pipelines were constructed using `albumentations` (v2.0.8) [4] and `torchvision` (v0.25.0). All hyperbolic manifold operations and distance calculations were performed using the `geoopt` library. Complete environment details and dependencies are provided in the `requirements.txt` file included in the supplementary code release.

Table 7: Embedding size and parameter counts on iWildCam. All models share the ConvNeXt-Tiny backbone (28.6M parameters). Each OOPark prototype occupies a single 20D sub-manifold, with its remaining coordinates parked at the origin. Training times are reported in Table 8; inference cost is dominated by the shared backbone, to which OOPark adds N small projection heads and N nearest-prototype searches.

Model	Total emb. dim	Head params	Total params
Euclidean 128	128	0.5	28.3
Euclidean 512	512	0.7	28.5
MGP 512	512	0.7	28.5
Single Lorentz 20	20	0.4	28.2
Single Lorentz 60	60	0.5M	28.3
OOPark (3×20)	60	1.2M	29.0M

Approximate wall-clock training times per seed are given in Table 8. No significant difference in training speed was observed between the compared methods over the full 100-epoch budget, and most runs converged before the final epoch.

Table 8: Approximate wall-clock training time per seed (100 epochs, one GPU), similar for all models.

Dataset	**Training time per seed**
Skin lesions (AISC)	5 hours
iWildCam	5 hours
Fungi	18 hours
Plankton	10 hours

Table 9 breaks down the compute for the experiments reported in this paper, approximately 1,500 GPU-hours in total. Runs shared between experiments are counted once. Post-hoc analyses (calibration, radius-stratified accuracy, gradient and curvature tracking) reuse stored predictions and training logs, and h-MDS prototype generation takes 1 minute per taxonomy, so neither adds meaningful compute. The full research project also required preliminary and discarded experiments that are not included in this estimate.

To ensure statistical robustness, all experiments were performed across **five independent seeds** per model configuration. While a fixed budget of 100 epochs was allocated for each run, the following observations regarding training dynamics were made:

- **Methodological Speed:** No significant difference in training speed was observed between the compared methods when evaluated over the full 100-epoch duration.
- **Convergence:** In most instances, model convergence was observed prior to the completion of the 100th epoch, indicating that the training duration was sufficient for reaching optimal performance.

Table 9: Approximate compute per reported experiment, in GPU-hours, derived from Table 8.

Experiment	Runs	GPU-hours
Main comparison (Table 2)	6 models $\times$ 4 benchmarks $\times$ 5 seeds	1,140
Parking ablation (Fig. 2a)	3 variants $\times$ 5 seeds (iWildCam)	75
Dimensionality ablation (Fig. 1a)	3 dimensions $\times$ 5 seeds (iWildCam)	75
Backbone study (Fig. 1b)	3 backbones $\times$ 2 models $\times$ 5 seeds (iWildCam)	150
Skin-lesion macro-split (Section D)	5 seeds	25
h-MDS stopping criterion (Fig. 8)	3 seeds	52
h-MDS sensitivity (Fig. 7)	9 configurations $\times$ 15 epochs (Plankton)	15
Total		$\approx 1{,}500$

F.5 Augmentations

To accommodate the diverse visual properties of our benchmarks, we employed two distinct data augmentation strategies. A standard crop-based pipeline (Table 10) was applied to the Skin Lesions, iWildCam, and Fungi datasets, whereas a specialized scale-and-pad pipeline (Table 11) was utilized exclusively for the Plankton dataset to preserve the extreme aspect ratios inherent to its imaging process.

Table 10: Data augmentation pipeline applied during the training and validation phases. The target resolution for all input images is 224 x 224 pixels. Prob denotes probability of applying it.

Phase	Augmentation	Parameters	Prob.
Training	Random Resized Crop	Scale: 0.7 to 1.0, Ratio: 0.8 to 1.2	1.0
	Transpose	None	0.5
	Vertical Flip	None	0.5
	Horizontal Flip	None	0.5
	Random Rotate (90°)	None	0.5
	Shift, Scale, Rotate	Shift: 5%, Scale: 5%, Rotate: 15°	0.5
	Morphological Distortion	Optical (limit: 0.5) *or* Grid (steps: 5, limit: 0.3)	0.3
	Color Jitter	Brightness/Contrast/Saturation: 0.2, Hue: 0.02	0.5
	Coarse Dropout	Max holes: 8, Max size: 28 x 28 pixels	0.3
	Normalization	Mean: (0.485, 0.456, 0.406), Std: (0.229, 0.224, 0.225)	1.0
Validation	Smallest Max Size	Shortest edge scaled to target size	1.0
	Center Crop	Size: 224 x 224 pixels	1.0
	Normalization	Mean: (0.485, 0.456, 0.406), Std: (0.229, 0.224, 0.225)	1.0

F.6 Distribution shift severity

We quantify the distribution shift severity for each benchmark below.

Table 11: Data augmentation pipeline applied to the Plankton dataset. To preserve extreme natural aspect ratios (e.g., 4:1) typical of micro-organisms, images are scaled by their longest edge and padded with a constant white background rather than cropped. Morphological distortions are strictly limited to respect the physical constraints of hard-shelled organisms. Prob denotes probability of applying it.

Phase	Augmentation	Parameters	Prob.
Training	Longest Max Size	Longest edge scaled to 224 pixels	1.0
	Constant Padding	Pad to 224 x 224 (Fill: White)	1.0
	Transpose	None	0.5
	Vertical Flip	None	0.5
	Horizontal Flip	None	0.5
	Random Rotate (90°)	None	0.5
	Shift, Scale, Rotate	Shift: 5%, Scale: 5%, Rotate: 15° (Fill: White)	0.5
	Morphological Distortion	Optical (limit: 0.2) *or* Grid (steps: 5, limit: 0.2)	0.3
	Color Jitter	Brightness/Contrast/Saturation: 0.2, Hue: 0.02	0.5
	Coarse Dropout	Max holes: 8, Max size: 28 x 28 pixels (Fill: White)	0.3
	Normalization	Mean: (0.485, 0.456, 0.406), Std: (0.229, 0.224, 0.225)	1.0
Validation	Longest Max Size	Longest edge scaled to 224 pixels	1.0
	Constant Padding	Pad to 224 x 224 (Fill: White)	1.0
	Normalization	Mean: (0.485, 0.456, 0.406), Std: (0.229, 0.224, 0.225)	1.0

Table 12: Validation versus test balanced accuracy for OOPark across the four benchmarks, illustrating intrinsic shift severity per setting. Drops Δ are computed as (Val $-$ Test). Validation splits are constructed in-distribution: a leaf-stratified 10% hold-out of AISC, the canonical WILDS `ood_val` split for iWildCam, the Danish-only subset of FungiTastic for Fungi, and a leaf-stratified 10% hold-out of WHOI for Plankton. The iWildCam row is reported for completeness but the WILDS protocol places both validation (`ood_val`) and test (`ood_test`) splits in OOD camera deployments relative to training, so the val–test gap there does not measure shift severity in the same sense as the other rows. All values in percent; $\pm$ denotes standard deviation across 5 random seeds.

Benchmark	Val L1	Test L1	ΔL1	Val Leaf	Test Leaf	ΔLeaf
ISIC	85.0 ± 3.0	68.5 ± 1.3	16.5	73.5 ± 4.0	59.7 ± 1.4	13.8
iWildCam[†]	71.0 ± 2.0	59.2 ± 1.8	11.8	50.0 ± 1.0	45.9 ± 1.4	4.1
Fungi	92.1 ± 2.0	91.5 ± 1.1	**0.6**	61.4 ± 1.6	58.2 ± 0.8	**3.2**
Plankton	75.7 ± 3.4	64.6 ± 2.8	11.0	54.7 ± 4.2	41.3 ± 3.3	13.4

[†]Both iWildCam validation and test splits are out-of-distribution with respect to the training cameras under the WILDS protocol; the val–test gap therefore does not isolate shift severity in the same way as the other benchmarks.

F.7 Calibration results

Table 13: Comparison of Pooled vs. L1 Macro-averaged metrics on Plankton.

	Micro-Averaged			L1 Macro-Averaged		
Model	**ECE** $\downarrow$	**MSP** $\uparrow$	**Radius** $\uparrow$	**ECE** $\downarrow$	**MSP** $\uparrow$	**Radius** $\uparrow$
Euclid. (512D)	0.35 ± 0.05	0.63 ± 0.06	—	0.21 ± 0.02	0.56 ± 0.02	—
MGP.(512D)	0.16 ± 0.05	0.72 ± 0.05	—	0.12 ± 0.02	**0.65** ± 0.04	—
S.Lorentz(20D)	**0.22** ± 0.06	0.68 ± 0.05	0.66 ± 0.04	**0.14** ± 0.03	0.52 ± 0.03	0.48 ± 0.03
S.Lorentz(80D)	0.32 ± 0.05	0.73 ± 0.02	0.72 ± 0.02	0.21 ± 0.02	0.54 ± 0.03	0.54 ± 0.03
OOPark(80D)	0.23 ± 0.03	**0.73** ± 0.01	0.71 ± 0.02	0.22 ± 0.03	0.59 ± 0.03	**0.56** ± 0.04

Table 14: Comparison of Pooled vs. L1 Macro-averaged metrics on Fungi

	Micro-Averaged			L1 Macro-Averaged		
Model	**ECE** $\downarrow$	**MSP** $\uparrow$	**Radius** $\uparrow$	**ECE** $\downarrow$	**MSP** $\uparrow$	**Radius** $\uparrow$
Euclid.(512D)	**0.05** $\pm$ 0.0	0.86 $\pm$ 0.0	—	**0.08** $\pm$ 0.0	0.88 $\pm$ 0.0	—
Euclid.(128D)	0.07 $\pm$ 0.1	0.86 $\pm$ 0.0	—	0.10 $\pm$ 0.0	0.87 $\pm$ 0.0	—
MGP (512D)	0.07 $\pm$ 0.00	0.65 $\pm$ 0.02	—	**0.08** $\pm$ 0.01	0.71 $\pm$ 0.00	—
S.Lorentz(20D)	0.11 $\pm$ 0.0	0.86 $\pm$ 0.0	0.80 $\pm$ 0.0	0.15 $\pm$ 0.0	**0.90** $\pm$ 0.0	**0.86** $\pm$ 0.0
S.Lorentz(100D)	0.11 $\pm$ 0.0	0.83 $\pm$ 0.0	0.78 $\pm$ 0.0	0.18 $\pm$ 0.01	0.89 $\pm$ 0.01	0.66 $\pm$ 0.01
OOPark(100D)	0.10 $\pm$ 0.0	**0.87** $\pm$ 0.0	**0.84** $\pm$ 0.0	0.12 $\pm$ 0.0	0.87 $\pm$ 0.0	0.85 $\pm$ 0.0

F.8 Temperature scaling for prototype based models

Since the cross entropy is regularized heavily in prototype based models, its common practice to perform temperature scaling on logits before reporting calibration metrics. See Table 15 for parameters.

Table 15: Temperature scaling hyperparameter sweep details.

Parameter Regime	**Search Range**	**Grid Points**
Sub-unit Temperature ($T < 1$)	$[0.05, 1.0)$	90
Super-unit Temperature ($T \geq 1$)	$[1.0, 10.0]$	90
Selection Criterion	Minimum Negative Log-Likelihood (NLL) on Validation Set	

F.9 Curvature results

Below is the inverse curvature of the product spaces, alongside the inverse curvature of the "global tree" for the single origin model. We note that the inverse curvature converges much earlier, and at far more curved setttings than the unparked model.

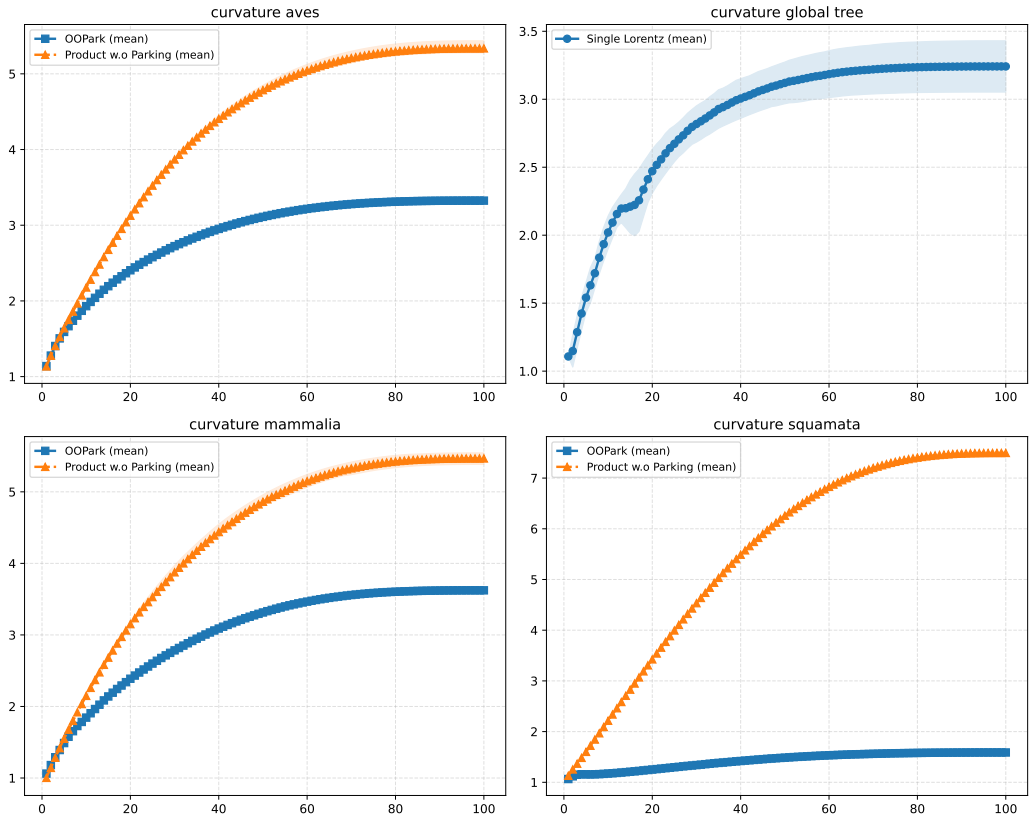


Figure 9: inverse curvature of OOPark, product space without parking and single Lorentz manifold

G Broader impact

OOPark aims to make classification under distribution shift more robust and to keep errors within the correct branch of a taxonomy. This is relevant to biodiversity monitoring, where camera-trap, fungal and plankton images are collected under conditions that differ from the training data, and to medical image triage, where a taxonomically close error (a related benign lesion) is less harmful than a distant one. The main risk is over-reliance: improved robustness on benchmarks does not imply reliability in deployment, and OOPark has not been validated for clinical use. Any medical application would require prospective clinical validation, evaluation across patient populations and skin types, and human oversight of predictions.

Table 16: Licenses and terms of use of the assets used in this work.

Asset	License / terms
AISC [48]	Not public; available on request
ISIC 2019 [7, 49]	CC BY-NC 4.0
iWildCam v2.0 (WILDS) [2, 22]	CC BY-NC-SA 4.0
Danish Fungi [39]	BSD 3-Clause License
FungiTastic [38]	BSD 3-Clause License
WHOI-Plankton [35]	MIT
FlowCamNet [19]	cc-by-nc-4.0
GBIF Backbone Taxonomy [11]	CC BY 4.0
WoRMS [55]	CC BY 4.0
PyTorch, torchvision	BSD-3-Clause
timm [53]	Apache-2.0
geoopt [21]	Apache-2.0
albumentations [4]	MIT

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 state the four-benchmark scope, the product-manifold architecture with origin parking, and the embedding-dimensionality claim (40–100D against up to 512D), which is explicitly not a parameter-count claim (Section F.4). The claim that OOPark generally outperforms the baselines is supported by Table 2, including a hierarchy-aware baseline (MGP) and a single manifold of matched dimensionality, and is qualified in the Discussion, which reports that OOPark trails the Euclidean baselines at the leaf level on Fungi (Section 6).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The "Limitations and future work" paragraph (Section 6) addresses the reliance on a static macro-taxonomy and fixed prototypes, the dependence on the h-MDS stopping criterion, sensitivity to annotations, the restriction to biological image benchmarks, and the scaling of the parking loss with the number of macro-branches N. The Discussion also states when the structural prior is not beneficial (mild shift, Fungi leaf level).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings,

model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper contains no theorems or formal results. The geometric arguments in Section 3 (angular crowding, the geometric equilibrium of the parking and contrastive terms) are explicitly framed as analytical intuitions that motivate the design rather than formal guarantees, and are tested empirically.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

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/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper documents the projection heads (Table 6, Section F.3), the data splits (Section F.1), the augmentations (Tables 10 and 11), the h-MDS prototype generation and its target distances (Section E), the long-tailed rebalancing (Section F.2), the tangent-vector clipping bound, the early-stopping criterion, post-hoc temperature scaling, and the configuration of every baseline, including the MGP regularization weight. Table 4 reports the optimizer, schedule, batch size, all learning rates and weight decays, and the loss and h-MDS constants. Code is released (see the next item).

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is available at `https://github.com/PeterKampen/OOPark`, with a README describing the environment, data preparation and the commands to reproduce the experiments. All evaluation datasets are publicly available from their original sources; the skin-lesion training set (AISC) is available from its authors on request but cannot be redistributed by us, so the skin-lesion results cannot be fully reproduced by third parties.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Data splits, including the canonical WILDS splits for iWildCam and the leaf-stratified 10% hold-outs, are given in Section F.1; the optimizer, per-component learning rates and weight decays, batch size and loss and h-MDS constants in Table 4; augmentations, rebalancing and early stopping in Section F and the Implementation Details. Hyperparameters were set in preliminary single-manifold experiments on the skin-lesion task and held fixed across all benchmarks and models; the MGP regularization weight was selected on validation. Sensitivity to the h-MDS constants is reported in Section E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Results are reported as mean and standard deviation ($\pm 1\sigma$) over five random seeds, which vary the initialization and data ordering (Section 5, Table 2); shaded bands and error bars in the figures show the same quantity. Analyses run with a single seed (radius-stratified accuracy, the h-MDS sensitivity sweep) are marked as such in their captions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section F.4 reports the hardware (NVIDIA L40S GPUs, under 24 GB VRAM per run), software versions, wall-clock training time per seed for each benchmark, and parameter counts for all models. The reported experiments required approximately 1,500 GPU-hours in total (Table 9); preliminary and discarded experiments required additional compute that is not included in this estimate.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is methodological and uses existing image datasets. All evaluation data are public research benchmarks; the skin-lesion images are anonymized, and the AISC training set [48] is covered by an ethical waiver from the Danish Ethics Committee (H-20066667) with access, anonymization and storage approved by the Danish Health Data Authorities and Data Protection Agency (21/5103, 18/53664). No new data were collected from people, and no crowdsourcing was involved. The authors have reviewed the NeurIPS Code of Ethics and the work conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

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 G discusses positive impacts (more robust classification for biodiversity monitoring and medical triage under distribution shift) and risks, in particular over-reliance on automated predictions in clinical settings, for which OOPark has not been validated.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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 (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases only training and evaluation code for a classification architecture; it releases no data, no generative model and no scraped dataset, and therefore poses no high-risk release scenario.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets, taxonomy services and software libraries are cited, and their licenses and terms of use are listed in Table 16. The AISC dataset is not publicly licensed; it is available from its authors on reasonable request [48] and is not redistributed.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

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 code, released at `https://github.com/PeterKampen/OOPark` under the MIT license, with a README documenting installation, data preparation and usage.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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 screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing and no research with human subjects; it uses only existing image datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No new data were collected from human subjects. The skin-lesion images are retrospective, anonymized clinical images from existing datasets; the AISC data [48] are covered by an ethical waiver from the Danish Ethics Committee (H-20066667) and approvals from the Danish Health Data Authorities and Data Protection Agency (21/5103, 18/53664).

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not part of the method or the experiments.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.