HyperTree: Scalable Unsupervised Hierarchy Discovery in Hyperbolic Space

Thomas Lang[⋆,1,2], **Kevin Sidak**[⋆,1,2], **Anna Beer**[3,1],
Sebastian Tschiatschek[1,4], **Yllka Velaj**[1,4,⋄], **Claudia Plant**[1,4,⋄]
[1] Faculty of Computer Science, University of Vienna, Vienna, Austria
[2] UniVie Doctoral School Computer Science, University of Vienna, Vienna, Austria
[3] Webster Vienna Private University, Vienna, Austria
[4] ds:UniVie, University of Vienna, Vienna, Austria
⋆ Joint first authors, ⋄ Shared supervision
firstname.lastname@univie.ac.at

Abstract

Real-world image datasets are deeply hierarchical, yet unsupervised hierarchical representation learning remains underdeveloped. Hyperbolic space is geometrically ideal for embedding trees with low distortion due to its exponential volume growth, but existing unsupervised hyperbolic methods either fail to scale beyond a few thousand points or depend heavily on class supervision. We bridge this gap with HyperTree, a scalable, capacity-efficient framework with a fully unsupervised objective, built on three contributions: (i) a geometrically consistent, closed-form definition of the hyperbolic Lowest Common Ancestor, valid across hyperbolic manifolds of arbitrary curvatures; (ii) an almost everywhere differentiable hierarchical objective based on Dasgupta's cost that is bounded and invariant to dataset scale; and (iii) a sample-complexity guarantee that enables scaling to datasets such as ImageNet-1k and iNaturalist-2018. Across four benchmarks, HyperTree outperforms Euclidean deep-clustering baselines by wide margins on hierarchical dendrogram purity and surpasses supervised hyperbolic baselines even when the baseline uses up to 8× more dimensions. **We release our code at https://github.com/univie-dm/HyperTree**.

1 Introduction

Real-world image datasets are deeply hierarchical. ImageNet's 1,000 classes nest within the WordNet taxonomy [Miller, 1995]; iNaturalist's 8,000+ species sit inside a biological one. Recovering such structure from unlabeled data is a longstanding goal of representation learning, and one where the choice of geometry is decisive. The hyperbolic space embeds trees with arbitrarily low distortion thanks to its exponential volume growth [Sala et al., 2018, Sarkar, 2011]; Euclidean space, the default for nearly all modern representation learning and deep clustering methods [Van Gansbeke et al., 2020, Chen et al., 2020, Grill et al., 2020, Caron et al., 2021, Qian, 2023], provably cannot [Matoušek, 1999]. The same volume growth also enables hyperbolic embeddings to capture rich hierarchies in very few dimensions, whereas Euclidean embeddings require many more (Figure 1).

Despite this geometric advantage, unsupervised hyperbolic hierarchical learning has not matured into a method that is simultaneously *label-free*, *scalable*, and *capacity-efficient at low dimension*. Existing approaches sacrifice at least one: they rely on class supervision [Yan et al., 2021, Kim et al., 2023], or they break down beyond a few thousand points [Monath et al., 2019, Chami et al., 2020]. Two obstacles underlie this gap. *Geometrically*, even defining the Lowest Common Ancestor (LCA) of two points in hyperbolic space — the basic primitive on which a hierarchical objective rests — has

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

lacked a closed form that is consistent across manifold models and handles the ancestor-descendant case correctly. *Computationally*, the natural quality measure for a learned tree, Dasgupta [2016]'s cost, scales cubically in the number of points n, confining prior continuous relaxations to a few thousand points or pushing them to lean on supervision to avoid this bottleneck [Monath et al., 2019, Chami et al., 2020, Lin et al., 2023].

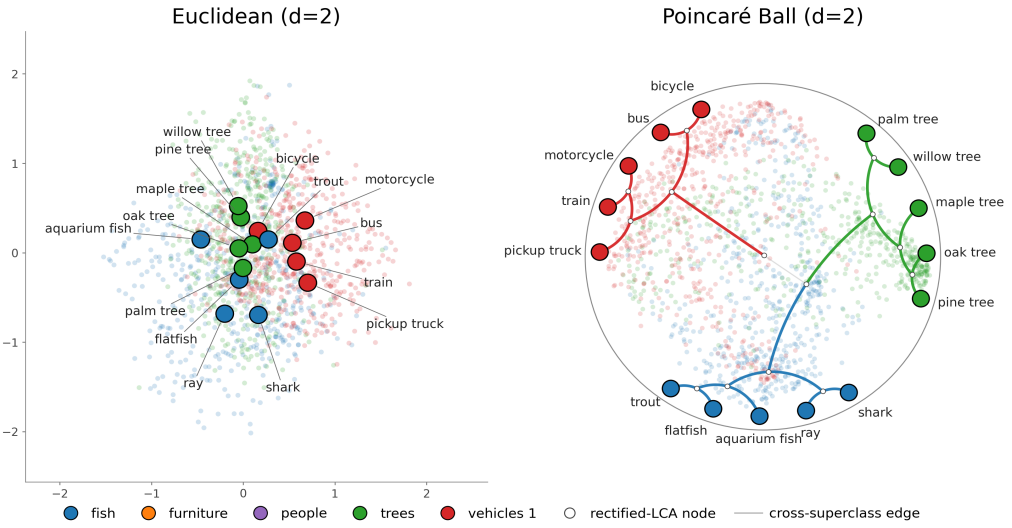

Figure 1: **Geometry is decisive at low dimension.** Two-dimensional embeddings of three CIFAR-100 superclasses (fish, trees, vehicles 1) together with their fine-grained classes. In Euclidean space (left), superclasses collapse into an indistinct cluster, while the Poincaré Ball under our framework (right) recovers a clean hierarchy. White markers denote the rectified hyperbolic LCAs (Section 4.2), and the gray edge joins all three superclasses at the root.

Our framework removes both obstacles, and we give the first geometrically consistent, closed-form definition of the hyperbolic LCA, valid on both the Poincaré Ball and the Hyperboloid at arbitrary curvature, with the ancestor-descendant case handled correctly. We extend Wang and Wang [2018]'s scale-normalized refinement of Dasgupta's cost to derive a hyperbolic objective that shares Dasgupta's minimizers while being differentiable almost everywhere, bounded in $[0, 1]$, invariant to dataset size, edge density, and weight scale. Finally, we prove a sample-complexity bound (Theorem 4.5) showing that this objective can be estimated to be within ε of its true value with probability $1 - \delta$ from a number of triplets that depends only on (ε, δ) and **not** on the size of the dataset n. The same fixed budget of $\approx 27{,}000$ triplets per step suffices for a 10^5-image dataset and a 10^7-image one, replacing the $\mathcal{O}(n^3)$ enumeration cost the exact Dasgupta loss would require. To our knowledge, this is the first such guarantee for a continuous unsupervised hierarchical objective, and it is what makes training on ImageNet-1k [Deng et al., 2009] and iNaturalist-2018 [Van Horn et al., 2018] feasible.

Our main contributions are:

1. **A rectified hyperbolic LCA.** A geometrically consistent, closed-form definition of the hyperbolic LCA, with the ancestor–descendant case handled correctly for the first time. Our derivation is valid for the Poincaré Ball and the Hyperboloid at arbitrary curvature, enabling practitioners to pick the model whose numerical and geometric trade-offs best fit their setting.
2. **A scale-invariant, almost everywhere differentiable hierarchical objective**, paired with the first sample-complexity guarantee for a differentiable hierarchical objective — enabling estimation at a cost independent of the dataset size.
3. **HyperTree, the first unsupervised deep hyperbolic hierarchical learning framework demonstrated at scale.** Training on ImageNet-1k and iNaturalist-2018 in as few as 16 latent dimensions, it sets a new state of the art in hierarchical dendrogram purity (hDP) [Heller and Ghahramani, 2005] among unsupervised methods across four benchmarks. Notably, it outperforms the supervised hyperbolic method HIER [Kim et al., 2023] on all benchmarks at matched dimensionality, and remains ahead even when HIER uses up to $8\times$ more dimension.

2 Hyperbolic Geometry

In this work, we follow standard notation and definitions for hyperbolic geometry [Cannon et al., 1997, Ratcliffe, 2006] and employ two models of hyperbolic space: the *Poincaré Ball* and the *Hyperboloid*. While both models capture the same underlying hyperbolic geometry, they provide distinct representations and offer unique views. They are isometric models and defined as simply-connected Riemannian manifolds $(\mathcal{M}_c^d, g^{\mathcal{M}_c^d})$ with constant negative sectional curvature $-c$, where $\mathcal{M}_c^d$ is a smooth manifold, $g^{\mathcal{M}_c^d}$ a smoothly varying inner product on its tangent spaces, and $c \in \mathbb{R}_{>0}$.

The d-dimensional **Poincaré Ball** $(\mathbb{P}_c^d, g^{\mathbb{P}_c^d})$ is defined as the open ball of radius $1/\sqrt{c}$ centered at the origin $\mathbb{P}_c^d = \left\{\boldsymbol{x} \in \mathbb{R}^d \colon\ c||\boldsymbol{x}||^2 < 1\right\}$ endowed with the Riemannian metric $g_{\boldsymbol{x}}^{\mathbb{P}_c^d} = (\lambda_{\boldsymbol{x}}^c)^2\, g_{\boldsymbol{x}}^{\mathbb{E}^d}$, where $\lambda_{\boldsymbol{x}}^c = 2/(1 - c\,||\boldsymbol{x}||^2)$ denotes the conformal factor and $g_{\boldsymbol{x}}^{\mathbb{E}^d}$ the Euclidean metric tensor. This conformal relationship to Euclidean space can also be expressed by the family of inner products $\left\{\langle \boldsymbol{u}, \boldsymbol{v}\rangle_{\boldsymbol{x}} \colon\ \mathcal{T}_{\boldsymbol{x}}\mathbb{P}_c^d \times \mathcal{T}_{\boldsymbol{x}}\mathbb{P}_c^d \to \mathbb{R},\ (\boldsymbol{u}, \boldsymbol{v}) \mapsto (\lambda_{\boldsymbol{x}}^c)^2 \langle \boldsymbol{u}, \boldsymbol{v}\rangle\right\}_{\boldsymbol{x} \in \mathbb{P}_c^d}$ induced by the metric. Here, $\mathcal{T}_{\boldsymbol{x}}\mathbb{P}_c^d$ is the tangent space with base point $\boldsymbol{x} \in \mathbb{P}_c^d$ and $\langle ., . \rangle$ the standard Euclidean dot product.

The d-dimensional **Hyperboloid** model $(\mathbb{H}_c^d, g^{\mathbb{H}_c^d})$ is defined as the forward sheet of a two-sheeted hyperboloid embedded within the $(d+1)$-dimensional Minkowski space $\mathbb{R}^{1,d}$ with $\mathbb{H}_c^d = \left\{(x_0, \ldots, x_d) \in \mathbb{R}^{d+1} \colon\ c\,\langle \boldsymbol{x}, \boldsymbol{x}\rangle_L = -1,\ x_0 > 0\right\}$. That is, the ambient space is the real vector space $\mathbb{R}^{d+1}$ endowed with the Minkowski inner product – an indefinite bilinear form defined as $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L = -x_0 y_0 + x_1 y_1 + \cdots + x_d y_d$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{d+1}$. The Riemannian metric on $\mathbb{H}_c^d$ is then obtained by restricting $\langle ., . \rangle_L$ to the tangent spaces of the Hyperboloid, yielding a positive-definite metric as required for a Riemannian manifold.

3 Related Work

Hyperbolic Representation Learning. Early work [Nickel and Kiela, 2017, 2018] showed that Poincaré Ball and Hyperboloid embeddings outperform Euclidean ones for hierarchical data, particularly in low dimensions. Both learn embeddings via a contrastive pairwise objective, leaving any hierarchy to emerge implicitly from the geometry rather than being optimized for. This implicit approach was later extended to graphs [Chamberlain et al., 2017] and images [Khrulkov et al., 2020]. More recently, Yang et al. [2023] used hyperbolic geometry purely as a regularizer, and Duan et al. [2025] used it to guide novel-class clustering with labeled subsets. Closer to our work are methods that train hyperbolic embeddings with an explicit hierarchical objective. Yan et al. [2021] learn hyperbolic embeddings with a log-ratio loss over data triplets targeting hierarchical clustering. Still, their pipeline depends on a fully supervised GoogLeNet backbone and is bottlenecked by the cubic cost of triplet enumeration, which limits their batch size to 80 - a regime incompatible with modern contrastive pretraining.[1] Lin et al. [2023] use hyperbolic autoencoders for multi-view clustering and target small datasets only, thus remaining unproven at ImageNet-1k scale.
Our closest architectural peer is HIER [Kim et al., 2023], which augments deep metric learning with a hyperbolic LCA regularizer. While both HIER and our method HyperTree organize hyperbolic embeddings via an LCA-based objective, they differ fundamentally: HIER requires full fine-grained labels and is restricted to the unit-curvature Poincaré Ball, whereas HyperTree uses a fully unsupervised joint objective and operates at arbitrary curvatures across multiple hyperbolic space models. We compare directly against HIER on hierarchical dendrogram purity in Section 5.

Hyperbolic Hierarchical Clustering. gHHC [Monath et al., 2019] and HypHC [Chami et al., 2020] are hierarchical clustering methods that optimize differentiable relaxations of discrete hierarchical costs in hyperbolic space. However, both have scalability bottlenecks. gHHC embeds internal tree nodes and caps their number a priori. Still, the joint optimization does not scale: their ImageNet-1k results train on a 50k-image subsample, with the remaining points assigned post-hoc via parent selection. HypHC instead optimizes only leaf embeddings, but relies on precomputed pairwise similarities over all n instances — an $\mathcal{O}(n^2)$ bottleneck that limits it to small datasets. Additionally, triplet enumeration costs $\mathcal{O}(B^3)$ per step, scaling poorly to the batch sizes contrastive training requires. Notably, both methods rely on the notion of hyperbolic LCAs. However, current definitions

[1] We could not evaluate their official implementation as a numerical baseline due to missing loss components and execution errors with standard backbones. Consequently, we treat their method strictly as related work.

in the literature are structurally inconsistent in certain configurations - a flaw we rectify in Section 4.2. Our approach systematically eliminates both bottlenecks. With our rectified hyperbolic LCA, the hierarchy emerges naturally from the geometry of the leaf embeddings alone, without maintaining additional parameters. We further replace the expensive evaluation of the relaxed hierarchical cost with an importance-rejection sampling strategy, backed by a sample-complexity bound (Theorem 4.5) whose required number of accepted triplets is independent of both n and B, enabling training on massive image datasets.

Deep Representation Learning and Clustering in Euclidean Space. Euclidean unsupervised methods form three groups [Zhou et al., 2025]: Autoencoder-based methods [Xie et al., 2016] learn embeddings by reconstructing the input, but the reconstruction objective is only loosely aligned with semantic structure and tends to preserve low-level appearance. Generative approaches based on VAEs or GANs [Kingma and Welling, 2013, Goodfellow et al., 2014] learn rich data distributions but similarly do not optimize for discriminability between semantically distinct instances. Contrastive methods [Chen et al., 2020, Grill et al., 2020, Caron et al., 2021] instead pull augmented views of the same image together while pushing different images apart. They are highly competitive on image benchmarks [Zhou et al., 2025], and serve as the standard backbone for deep clustering pipelines. SCAN [Van Gansbeke et al., 2020] and SeCu [Qian, 2023] exemplify this trajectory: both pair contrastive pretraining with a clustering-discrimination stage, and achieve strong results on standard flat clustering label-agreement metrics. However, neither exposes a hierarchy: the output is a single flat clustering at a pre-specified k. We include both as among the strongest unsupervised Euclidean baselines for comparison. HyperTree substantially outperforms both methods in hierarchical dendrogram purity on all three shared benchmarks (Section 5) and also achieves higher 1-NN accuracy on CIFAR-100 and TinyImageNet (Appendix A.7.5).

Comparison Summary. We compare our method to prior work along five axes in Table 1: operation in hyperbolic space, hierarchy-awareness (producing or directly using a hierarchical structure), unsupervised training, demonstration at scale ($\geq$100k images), and support for multiple hyperbolic manifolds.

Table 1: Comparison of representative methods. "Unsupervised training": ✓ if the training objective does not use any class labels; backbone supervision is indicated separately below. "Scaled to $\geq$100k images": ✓ only if the reported image experiments train on at least 100k images; ✗ does not preclude architectural scalability. "Multiple manifolds": support for more than one hyperbolic manifold.

Method	**Hyperbolic space**	**Hierarchy-aware**	**Unsupervised training**	**Scaled to $\geq$100k images**	**Multiple manifolds**
Nickel and Kiela [2017, 2018]	✓	✓	✓	✗†	✓
Khrulkov et al. [2020]	✓	✗	✗	✗	✗
Yan et al. [2021]	✓	✓	✓‡	✗	✗
Kim et al. [2023] (HIER)	✓	✓	✗	✓	✗
Monath et al. [2019] (gHHC)	✓	✓	✓‡	✗††	✗
Chami et al. [2020] (HypHC)	✓	✓	✓‡	✗	✗
Van Gansbeke et al. [2020] (SCAN)	✗	✗	✓	✓	N/A
Qian [2023] (SeCu)	✗	✗	✓	✓	N/A
Ours (HyperTree)	✓	✓	✓*	✓	✓

† Evaluated only on graph/taxonomy data; no image experiments reported.
‡ The training loss itself is unsupervised, but image experiments rely on supervised features.
†† ImageNet-1k result trains on a 50k subset; remaining points are post-hoc assigned to the pretrained tree.
* HyperTree uses no target labels. Only the iNaturalist-2018 backbone was pretrained on taxonomic-text-paired images.

No prior method combines an explicit, geometrically consistent hierarchical objective with a fully unsupervised joint objective in a curvature-agnostic framework that scales to ImageNet-1k and iNaturalist-2018 — the gap our method fills.

4 Hierarchical Representation Learning

To learn hierarchical representations at multiple scales in a low-dimensional hyperbolic space, we train a hybrid Euclidean-hyperbolic neural network. A standard Euclidean backbone extracts initial

features, which a Euclidean projection head progressively reduces in dimensionality. To bound gradients of the subsequent hyperbolic layers, we apply the feature scaling of Klein et al. [2025]. The rescaled features are then mapped into a hyperbolic space $(\mathcal{M}_c^d, g^{\mathcal{M}_c^d})$ by treating them as tangent vectors at the origin $\bar{\mathbf{0}}$ and applying the exponential map $\exp_{\bar{\mathbf{0}}} : \mathcal{T}_{\bar{\mathbf{0}}}\mathcal{M}_c^d \to \mathcal{M}_c^d$. A final hyperbolic linear layer produces the d-dimensional embeddings, following the Poincaré Ball formulation of Shimizu et al. [2021] or the Hyperboloid formulation of van der Klis et al. [2026]. Appendix A.2 details these operations and advanced layers and Figure 2 summarizes the architecture.

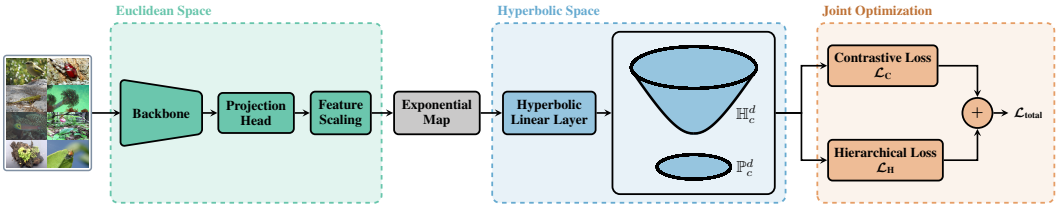

Figure 2: Hybrid model architecture. Euclidean features are mapped to the hyperbolic space via the exponential map and a hyperbolic linear layer for joint contrastive and hierarchical optimization.

4.1 Loss Function

The model is trained by jointly optimizing a contrastive loss $\mathcal{L}_C$ and our novel hierarchical loss $\mathcal{L}_H$:

$$\mathcal{L}_{\text{total}} = \lambda_C \mathcal{L}_C + \lambda_H \mathcal{L}_H, \qquad \lambda_C, \lambda_H \in \mathbb{R}_{>0}.$$

We first pretrain on $\mathcal{L}_C$ alone to establish high-quality latent representations, then linearly increase λ_H to progressively impose the multiscale hierarchical structure on the embedding space.

Contrastive Loss. We adopt the InfoNCE objective [Oord et al., 2018] with negative geodesic distance as similarity function, $\text{sim}(\cdot,\cdot) = -d_{\mathcal{M}_c^d}(\cdot,\cdot)$. The full formulation is in Appendix A.3.1.

Hierarchical Loss. Traditionally, the quality of a hierarchy T over n points with pairwise similarities is measured by Dasgupta [2016]'s cost. This cost, however, scales as $\mathcal{O}(n^3)$ for dense graphs and is highly sensitive to edge-weight magnitude, making it difficult to optimize and to compare across graphs of different size or density. Wang and Wang [2018] addressed this in the discrete setting via normalization, yielding a bounded ratio that reflects hierarchical structure independent of graph size. Yet evaluating it still requires enumerating all triplets, an $\mathcal{O}(n^3)$ operation. Differentiable relaxations [Chami et al., 2020] replace discrete LCA operations with continuous hyperbolic analogues, but inherit the magnitude sensitivity of the original Dasgupta cost on top of the cubic enumeration cost.

We propose $\mathcal{L}_H$, which carries the topology invariance of Wang and Wang [2018]'s discrete cost into an almost everywhere differentiable objective and, crucially, admits an unbiased estimator with sample complexity independent of n (Section 4.3):

$$\mathcal{L}_H(\boldsymbol{Z}) = \frac{\sum_t (N_t - N_t^{\min})}{\sum_t (N_t^{\max} - N_t^{\min})},$$

where $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ are the hyperbolic embeddings and $t = (i,j,k)$ ranges over unordered triplets of distinct points. Assuming a positive denominator, we define

$$N_t = w_{ij} + w_{ik} + w_{jk} - \left\langle \begin{pmatrix} w_{ij} \\ w_{ik} \\ w_{jk} \end{pmatrix}, \sigma_\tau \begin{pmatrix} d_{\mathcal{M}_c^d}(\bar{\mathbf{0}}, \text{LCA}(\boldsymbol{z}_i, \boldsymbol{z}_j)) \\ d_{\mathcal{M}_c^d}(\bar{\mathbf{0}}, \text{LCA}(\boldsymbol{z}_i, \boldsymbol{z}_k)) \\ d_{\mathcal{M}_c^d}(\bar{\mathbf{0}}, \text{LCA}(\boldsymbol{z}_j, \boldsymbol{z}_k)) \end{pmatrix} \right\rangle.$$

The terms $N_t^{\min}$, $N_t^{\max}$ are defined as the hard minimum and maximum of $\{w_{ij} + w_{ik}, w_{ij} + w_{jk}, w_{ik} + w_{jk}\}$, bound N_t from below and above (Proposition A.2), and are extrema of three smooth pairwise sums that admit subgradients everywhere. That is, they are differentiable except on the measure-zero set of ties and softening them would invalidate Theorem 4.5 (see Appendix A.3.5). The remaining components are the hyperbolic Lowest Common Ancestor $\text{LCA}(\cdot,\cdot)$, the temperature-scaled softmax function σ_τ, and the pairwise similarity $w_{ij} = 1/(1 + d_{\mathcal{M}_c^d}(\boldsymbol{z}_i, \boldsymbol{z}_j)) \in (0,1]$, rewarding proximity disproportionately.

By construction, $\mathcal{L}_{\mathrm{H}}$ is differentiable almost everywhere, bounded in $[0, 1]$, and invariant to the number of embedded points n (node count), edge density, and weight scaling. Its bounds make $\mathcal{L}_{\mathrm{H}}$ directly interpretable as a measure of hierarchical fit: star-like cluster structures, in which similar points share an early common ancestor near the leaves, yield values near 0, while flat or path-like arrangements receive proportionally higher values. Dasgupta [2016], Chami et al. [2020] costs, by contrast, assign an $\mathcal{O}(n^3)$ value to a dense unit-weight graph regardless of its underlying structure, and rescaling the edge weights inflates this cost arbitrarily; normalization corrects both issues. The resulting form exposes $\mathcal{L}_{\mathrm{H}}$ as a ratio of expectations under uniform triplet sampling, making it amenable to being an unbiased estimator that breaks the $\mathcal{O}(n^3)$ enumeration barrier shared by all prior Dasgupta- and Wang-based formulations (Section 4.3).

Consistency with discrete and continuous Dasgupta costs. Our normalization builds on the same triplet lower bound that underlies the base cost of Wang and Wang [2018]. However, instead of dividing by the base cost, we subtract this bound and divide by the attainable range. To relate our objective to the continuous relaxation of Dasgupta's cost proposed by Chami et al. [2020], we substitute our rectified LCA depths into their softmax formulation and write $C_{\mathrm{HypHC}}(\boldsymbol{Z}; w, \tau) = \sum_t N_t + 2\sum_{i<j} w_{ij}$. For fixed similarities, our normalization then preserves the minimizers of this relaxation:

Proposition 4.1 (Minimizer Equivalence under Fixed Similarities)**.** *Let the pairwise similarities $w = (w_{ij})$ be fixed, with $D(w) = \sum_t (N_t^{\max} - N_t^{\min}) > 0$, and let $\tau > 0$. Then $\mathcal{L}_H(\,\cdot\,; w)$ is a strictly increasing affine transformation of $C_{\mathrm{HypHC}}(\,\cdot\,; w, \tau)$, and the two objectives have the same minimizers over any embedding domain.*

Proof. See Appendix A.4.3. □

For fixed similarities, the normalization therefore preserves both the minimizers and the descent directions of the continuous relaxation of Dasgupta's cost. During joint training, the similarities change, but the correspondence still holds for the gradient with respect to the LCA depths at the current similarities: the gradient pushes more similar pairs toward deeper LCAs (Appendix A.4.3). In Section 5, we evaluate the hierarchies learned when this signal is combined with simultaneous updates to the similarities.

4.2 Hyperbolic Lowest Common Ancestors

Evaluating $\mathcal{L}_{\mathrm{H}}$ relies on the precise geometric formulation of the hyperbolic LCA. The standard definition [Monath et al., 2019, Chami et al., 2020, Lin et al., 2023, Duan et al., 2025] defines the hyperbolic LCA of $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}_c^d$ as the point $\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}})$ that is closest to the origin $\bar{\mathbf{0}}$ along their (indefinite) connecting geodesic $\Gamma_{\boldsymbol{x},\boldsymbol{y}}$. This fails when one point is an ancestor of the other: the resulting LCA falls outside the geodesic *segment* connecting them, violating the basic tree property that an LCA must lie on the shortest path between its descendants (Figure 3).

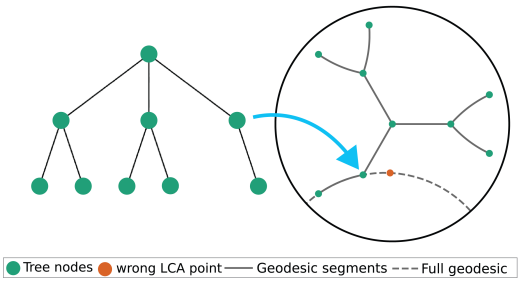

Figure 3: Illustration of the flaw in the previous hyperbolic LCA definitions. The "wrong LCA point" (orange) lies on the full geodesic (dashed line) but outside the segment (solid line) connecting its descendants — an inconsistency our rectified definition resolves.

We propose a rectified definition that constrains the LCA to the geodesic segment $\gamma_{\boldsymbol{x},\boldsymbol{y}} \subset \Gamma_{\boldsymbol{x},\boldsymbol{y}}$.

Definition 4.2. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}_c^d$ and $\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}})$ denote the orthogonal projection of the origin $\bar{\mathbf{0}}$ onto the full geodesic $\Gamma_{\boldsymbol{x},\boldsymbol{y}}$. The hyperbolic lowest common ancestor $\mathrm{LCA}(\boldsymbol{x}, \boldsymbol{y})$ is:

$$\mathrm{LCA}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \bar{\mathbf{0}}, & \text{if } \bar{\mathbf{0}} \in \gamma_{\boldsymbol{x},\boldsymbol{y}} \\ \arg\min_{\boldsymbol{z} \in \{\boldsymbol{x},\boldsymbol{y}\}} d_{\mathcal{M}_c^d}(\bar{\mathbf{0}}, \boldsymbol{z}), & \text{if } \bar{\mathbf{0}} \notin \gamma_{\boldsymbol{x},\boldsymbol{y}} \text{ and } \pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) \notin \gamma_{\boldsymbol{x},\boldsymbol{y}} \\ \pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}), & \text{otherwise.} \end{cases}$$

The first case handles antipodal points, which are excluded by the collinearity assumptions in Proposition 4.3. The second handles projections outside the segment and returns the endpoint closer to the origin. The third is the generic case in which the orthogonal projection of the origin onto $\Gamma_{\boldsymbol{x},\boldsymbol{y}}$ falls within the connecting segment: the rectified LCA then coincides with the geodesic orthogonal projection $\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}})$, the point along the segment closest to the origin. Both case distinctions use a constant number of inner products (see Appendix A.3.2).

To our knowledge, only restricted instances have been derived previously. Chami et al. [2020], for example, give a formula for the 2d Poincaré Disk at fixed curvature. The following two propositions close this gap, providing closed-form expressions in arbitrary dimension and at arbitrary curvature c for both the Poincaré Ball and, for the first time, the Hyperboloid.

Proposition 4.3. *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{P}_c^d$ be non-collinear. Then*

$$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) = \left(1 - \frac{\sqrt{||\boldsymbol{z}||^2 - \frac{1}{c}}}{||\boldsymbol{z}||}\right)\boldsymbol{z},$$

$$\boldsymbol{z} = \frac{\left[(1 + c||\boldsymbol{y}||^2)\langle \boldsymbol{x}, \boldsymbol{y}\rangle - (1 + c||\boldsymbol{x}||^2)||\boldsymbol{y}||^2\right]\boldsymbol{x} + \left[(1 + c||\boldsymbol{x}||^2)\langle \boldsymbol{x}, \boldsymbol{y}\rangle - (1 + c||\boldsymbol{y}||^2)||\boldsymbol{x}||^2\right]\boldsymbol{y}}{2c(\langle \boldsymbol{x}, \boldsymbol{y}\rangle^2 - ||\boldsymbol{x}||^2||\boldsymbol{y}||^2)}.$$

Proof. See Appendix A.4.1. □

Proposition 4.4. *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}_c^d$ with $x \neq y$. Then*

$$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) = \frac{\alpha\boldsymbol{x} + \beta\boldsymbol{y}}{\sqrt{-c\,\langle \alpha\boldsymbol{x} + \beta\boldsymbol{y}, \alpha\boldsymbol{x} + \beta\boldsymbol{y}\rangle_L}},$$

$$\alpha = \frac{\sqrt{c}\,(x_0 + c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L\, y_0)}{1 - (c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L)^2}, \qquad \beta = \frac{\sqrt{c}\,(y_0 + c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L\, x_0)}{1 - (c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L)^2}.$$

Proof. See Appendix A.4.1. □

4.3 Scalability and Sampling Complexity

Exact evaluation of $\mathcal{L}_{\mathrm{H}}$ requires enumerating all triplets, an $\mathcal{O}(n^3)$ cost that is infeasible at scale. We show that $\mathcal{L}_{\mathrm{H}}$ admits an unbiased estimator whose sample complexity is *independent of* n (or B if batches of size B are used instead), and derive an explicit finite-sample bound.
The key observation is that $\mathcal{L}_{\mathrm{H}}$ can be rewritten as a ratio of expectations under uniform sampling:

$$\mathcal{L}_{\mathrm{H}}(\boldsymbol{Z}) = \frac{\sum_t (N_t - N_t^{\min})}{\sum_t (N_t^{\max} - N_t^{\min})} = \frac{\mathbb{E}_{t\sim\mathcal{U}}\left[N_t - N_t^{\min}\right]}{\mathbb{E}_{t\sim\mathcal{U}}\left[N_t^{\max} - N_t^{\min}\right]}.$$

We estimate this ratio via importance sampling under the target distribution $P(t) \propto N_t^{\max} - N_t^{\min}$:

$$\mathcal{L}_{\mathrm{H}}^m(\boldsymbol{Z}) = \frac{1}{m}\sum_{l=1}^{m} \frac{N_{t_l} - N_{t_l}^{\min}}{N_{t_l}^{\max} - N_{t_l}^{\min}}, \qquad \text{with } t_l \sim P.$$

Sampling from P would normally require its $\mathcal{O}(n^3)$ normalizing constant. We sidestep this with a rejection-sampling scheme that draws triplets uniformly from the current batch and accepts them with probability $A(t) = (N_t^{\max} - N_t^{\min})/(w_{\max} - w_{\min})$, where $w_{\min}$ and $w_{\max}$ are the smallest and largest pairwise similarities of that batch. By construction, $N_t^{\max} - N_t^{\min} \leq w_{\max} - w_{\min}$, ensuring that $A(t) \leq 1$. The batch extrema act as exact normalizers and are used exclusively for this selection step, which is gradient-free: rejected candidates are evaluated from detached similarity lookups and never enter the backward pass. Thus, the required sample size is m accepted triplets. The full derivation is given in Appendix A.3.3 and Appendix A.3.4 discusses the batch-level normalizers.

The following theorem establishes a sample size sufficient for an (ε, δ)-accurate estimate.

Theorem 4.5 (Minimum Sample Complexity)**.** *For any $\varepsilon, \delta \in (0, 1)$, drawing*

$$m \geq \frac{\left(\frac{1}{2} + \frac{2}{3}\varepsilon\right)\ln\left(\frac{2}{\delta}\right)}{\varepsilon^2}$$

triplets from $P(t) \propto N_t^{\max} - N_t^{\min}$ guarantees $\mathbb{P}\left[|\mathcal{L}_H^m(\boldsymbol{Z}) - \mathcal{L}_H(\boldsymbol{Z})| \geq \varepsilon\right] \leq \delta$.

Proof. See Appendix A.4.2. □

This bound is particularly powerful as it depends only on the target precision ε and confidence $1-\delta$, **not** on n. For instance, achieving $\varepsilon = 0.01$ and $\delta = 0.01$ requires only $m \approx 27{,}000$ triplets, entirely circumventing the $\mathcal{O}(n^3)$ enumeration barrier. To our knowledge, this is the first sample-complexity guarantee for a differentiable hierarchical objective: Chami et al. [2020] fix their sampling rate empirically at $\mathcal{O}(n^2)$, and Monath et al. [2019] provide no comparable analysis.

5 Experiments

5.1 Setup

We evaluate HyperTree on four image-classification benchmarks with different class counts and hierarchy depths: CIFAR-100 [Krizhevsky et al., 2009] (100 classes; 8-level WordNet hierarchy), TinyImageNet [Le et al., 2015] (200 classes; 7 levels), ImageNet-1k [Deng et al., 2009] (1,000 classes; 8 levels), and iNaturalist-2018 [Van Horn et al., 2018] (8,142 species; 7-level biological taxonomy). On CIFAR-100 and TinyImageNet, we first pretrain a ResNet-50 [He et al., 2016] backbone and a Euclidean projection head with SimCLR [Chen et al., 2020]. We then train the backbone, projection head, feature scaling, and hyperbolic linear layer jointly with the contrastive and hierarchical objectives. On ImageNet-1k and iNaturalist-2018, we keep the backbone frozen and update only the projection head, feature scaling, and hyperbolic linear layer. The backbone is a DINO-pretrained ResNet-50 [Caron et al., 2021] for ImageNet-1k and the ViT-B/16 image encoder of BioCLIP [Stevens et al., 2024] for iNaturalist-2018. Our joint training does not use any target class labels. Further dataset and implementation details are provided in Appendices A.5.1 and A.6.

5.2 Evaluation and Baselines

Our primary evaluation metric is hierarchical dendrogram purity (hDP), which measures how well a predicted tree agrees with the ground-truth hierarchy [Heller and Ghahramani, 2005]. We compare HyperTree against two unsupervised Euclidean deep-clustering methods, SCAN [Van Gansbeke et al., 2020] and SeCu [Qian, 2023], and against HIER [Kim et al., 2023], a supervised hyperbolic method trained with fine-class labels. Baseline embeddings are extracted from the released SCAN and SeCu checkpoints and from our own rerun of HIER. All methods are then evaluated with the same pipeline.

For comparisons across methods, we construct dendrograms under Ward linkage on pairwise geodesic distances (Euclidean distances for the Euclidean baselines), which yields a common decoder for HyperTree and all external baselines. However, on non-Euclidean distances, Ward linkage generally loses its usual interpretation as minimizing within-cluster Euclidean variance. We therefore treat it purely as a fixed decoding procedure shared by all methods. For HyperTree, we additionally evaluate hDP on the *LCA-induced tree*. This tree is built by single-linkage agglomeration with rectified-LCA depths as similarities, where greater depth means greater similarity. Single linkage is part of this tree's definition, not a decoder choice tuned for performance. Since this decoder operates directly on the LCA depths that enter $\mathcal{L}_{\mathrm{H}}$, the resulting tree provides the most direct readout of the learned hierarchy. We therefore use it to compare variants of HyperTree and to measure improvement over the epoch-0 anchor, i.e. the model at the start of joint training. We examine decoder sensitivity in Section 5.4. Evaluation splits, metric definitions, decoding details, and baseline implementations are described in Appendices A.5, A.5.2 and A.6.5.

5.3 Hierarchical Clustering Results

Table 2 reports hDP for all methods under Ward linkage and, for HyperTree, also under the LCA-induced decoder. We define $\Delta_{\mathrm{LCA}} = h\mathrm{DP}_{\mathrm{LCA}}^{\mathrm{trained}} - h\mathrm{DP}_{\mathrm{LCA}}^{\mathrm{epoch}\,0}$ as the change in LCA-induced tree hDP from the corresponding epoch-0 model to the trained model.

5.3.1 Comparison with External Baselines

Under the shared Ward decoder, HyperTree outperforms SCAN, SeCu, and HIER on all three datasets for which baseline results are available. This includes HIER, which is trained with fine-class labels, whereas our joint training uses no target class labels.

Dataset	Method	Unsup.	d	$h\mathrm{DP}_{\mathrm{Ward}}$	$h\mathrm{DP}_{\mathrm{LCA}}$	Δ_{LCA}
CIFAR-100	SCAN	✓	128	0.353	n/a	n/a
	SeCu	✓	16	0.416 ± 0.011	n/a	n/a
	HIER	✗	16	0.797 ± 0.025	n/a	n/a
	HyperTree ($\mathbb{H}^d_c$)	✓	16	0.875 ± 0.017	0.891 ± 0.009	$+0.170 \pm 0.014$
	HyperTree ($\mathbb{P}^d_c$)	✓	16	$\mathbf{0.898 \pm 0.008}$	$\mathbf{0.912 \pm 0.007}$	$+0.173 \pm 0.011$
TinyImageNet	SCAN	✓	128	0.091 ± 0.003	n/a	n/a
	SeCu	✓	128	0.121 ± 0.028	n/a	n/a
	HIER	✗	16	0.447	n/a	n/a
	HyperTree ($\mathbb{H}^d_c$)	✓	16	0.849 ± 0.012	0.846 ± 0.018	$+0.026 \pm 0.021$
	HyperTree ($\mathbb{P}^d_c$)	✓	16	$\mathbf{0.862 \pm 0.006}$	$\mathbf{0.860 \pm 0.008}$	$+0.046 \pm 0.006$
ImageNet-1k	SCAN	✓	128	0.276	n/a	n/a
	SeCu	✓	128	0.334	n/a	n/a
	HIER	✗	32	0.531 ± 0.004	n/a	n/a
	HyperTree ($\mathbb{H}^d_c$)	✓	32	0.927 ± 0.013	0.928 ± 0.013	$+0.088 \pm 0.014$
	HyperTree ($\mathbb{P}^d_c$)	✓	32	$\mathbf{0.934 \pm 0.000}$	$\mathbf{0.936 \pm 0.001}$	$+0.097 \pm 0.003$
iNaturalist-2018	HyperTree ($\mathbb{H}^d_c$)	✓	128	0.918 ± 0.011	0.870 ± 0.040	$+0.100 \pm 0.039$
	HyperTree ($\mathbb{P}^d_c$)	✓	128	$\mathbf{0.921 \pm 0.009}$	$\mathbf{0.883 \pm 0.007}$	$+0.111 \pm 0.007$

Table 2: Hierarchical dendrogram purity ($h\mathrm{DP} \in [0, 1]$; higher is better) under Ward linkage and LCA-induced decoding for the Poincaré Ball ($\mathbb{P}^d_c$) and the Hyperboloid ($\mathbb{H}^d_c$). Here, d denotes the embedding dimension, and Δ_{LCA} the gain over the seed-matched epoch-0 model. LCA results are reported only for HyperTree; external comparisons use Ward linkage (n/a). "Unsup." denotes task-training with unsupervised objectives. We report mean $\pm$ standard deviation over three seeds where available and single-seed results otherwise. **Bold** marks the best result per dataset and decoder.

5.3.2 Training Gains on the LCA-Induced Tree

For both hyperbolic space models, the mean Δ_{LCA} is positive on all four datasets. Under the LCA-induced tree decoder, joint training improves hierarchy recovery over the matched epoch-0 representation. The gain is largest on CIFAR-100 and smallest on TinyImageNet. On iNaturalist-2018, which has the largest number of leaf classes, joint training improves LCA-induced hDP by $+0.100$ (Hyperboloid) and $+0.111$ (Poincaré Ball).

5.3.3 Poincaré Ball versus Hyperboloid

Since our rectified LCA is defined for both hyperbolic models, we compare them at matched intrinsic dimension and curvature, with otherwise identical training recipes (Table 2). The Poincaré Ball achieves higher mean hDP under both decoders on all four datasets and varies less across seeds. In some settings, however, the difference is fairly small relative to seed variability, most notably under Ward linkage on iNaturalist-2018.

5.4 Effect of Tree Decoding

To assess the effect of tree decoding independently of representation learning, Table 3 compares single, average, complete, and Ward linkage applied to geodesic distances, using the same trained embeddings for each hyperbolic space model.

On CIFAR-100, TinyImageNet, and ImageNet-1k, the Poincaré Ball achieves similarly high hDP under all geodesic-distance linkages and the LCA-induced tree (Table 2), indicating that the recovered hierarchy is largely decoder-insensitive. Ward linkage is also not consistently the best decoder on these datasets, so the cross-method comparison does not systematically rely on the decoder most favorable to HyperTree. On iNaturalist-2018, the decoder matters more: average and Ward linkage outperform both geodesic single linkage and the LCA-induced tree. Since all trees are decoded from the same embeddings, these gaps reflect the decoding procedure rather than the learned representation.

Dataset	Space	Single	Average	Complete	Ward
CIFAR-100	$\mathbb{P}_c^d$	0.911 ± 0.006	0.909 ± 0.008	0.907 ± 0.009	0.898 ± 0.008
	$\mathbb{H}_c^d$	0.889 ± 0.007	0.890 ± 0.009	0.875 ± 0.017	0.875 ± 0.017
TinyImageNet	$\mathbb{P}_c^d$	0.855 ± 0.005	0.863 ± 0.006	0.864 ± 0.005	0.862 ± 0.006
	$\mathbb{H}_c^d$	0.846 ± 0.017	0.851 ± 0.016	0.848 ± 0.015	0.849 ± 0.012
ImageNet-1k	$\mathbb{P}_c^d$	0.934 ± 0.001	0.937 ± 0.002	0.935 ± 0.001	0.934 ± 0.000
	$\mathbb{H}_c^d$	0.927 ± 0.013	0.930 ± 0.015	0.928 ± 0.014	0.927 ± 0.013
iNaturalist-2018	$\mathbb{P}_c^d$	0.880 ± 0.010	0.913 ± 0.002	0.911 ± 0.007	0.921 ± 0.009
	$\mathbb{H}_c^d$	0.857 ± 0.031	0.914 ± 0.006	0.910 ± 0.002	0.918 ± 0.011

Table 3: Decoder ablation: hDP ($\in [0, 1]$; higher is better) of the Poincaré Ball ($\mathbb{P}_c^d$) and Hyperboloid ($\mathbb{H}_c^d$) embeddings from Table 2, with the same embedding dimensions and curvatures, under different tree decoders. All linkage decoders operate on pairwise geodesic distances. We report mean $\pm$ standard deviation over three seeds.

5.5 Ablation Studies

We report further ablations in the Appendix. Appendix A.7.1 examines training dynamics and shows that much of the improvement in hierarchical purity occurs early in training. Appendix A.7.2 isolates the contribution of each loss component on CIFAR-100, TinyImageNet, and ImageNet-1k, where the combined objective outperforms either term alone. Appendix A.7.3 shows that larger embedding dimensions do not consistently improve performance, indicating that compact embeddings suffice for HyperTree. Appendix A.7.4 breaks down Ward linkage hDP by hierarchy level to show at which depths our gains arise. Appendix A.7.5 reports fine-class nearest-neighbor accuracy and flat clustering results, and Appendix A.7.6 examines the choice of curvature on iNaturalist-2018. Together, these ablations provide complementary views of the learned structure.

6 Conclusion

A compact image embedding can capture both what two images have in common and how general that common ground is. In HyperTree, the depth of their lowest common ancestor gives this generality a geometric meaning, making it a direct target for learning. On all four benchmarks, training on sampled triplets improved the agreement between the induced trees and the reference taxonomies, even though the objective uses neither target labels nor the ground-truth hierarchy. More broadly, these results show that a representation can learn to preserve relationships at several levels of granularity without ever seeing the taxonomy that defines them.

Limitations and Future Work. Our largest experiments used frozen backbones, leaving open whether end-to-end training would improve hierarchy recovery at that scale. On iNaturalist-2018, Ward linkage and the LCA-induced tree performed best at different curvatures. Tracing where the two decoders disagree as curvature varies could reveal which branches are most sensitive to the choice of decoder and inform how curvature is selected.

Acknowledgments and Disclosure of Funding

We thank the open-source community for providing the tools that enabled this work: PyTorch [Paszke et al., 2019], NumPy [Harris et al., 2020] and Matplotlib [Hunter, 2007]. The computational results have been achieved using the Austrian Scientific Computing (ASC) infrastructure. Lastly, we want to thank the anonymous reviewers who helped improve our work through their feedback.

References

Sergei Bernstein. On a modification of chebyshev's inequality and of the error formula of laplace. *Ann. Sci. Inst. Sav. Ukraine, Sect. Math*, 1(4):38–49, 1924.

Rajendra Bhatia and Chandler Davis. A better bound on the variance. *The american mathematical monthly*, 107(4):353–357, 2000.

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

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9630–9640. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00951.

Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*, 2017.

Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. *Advances in Neural Information Processing Systems*, 33:15065–15076, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. URL `https://dl.acm.org/doi/abs/10.5555/3524938.3525087`.

Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 118–127, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Yu Duan, Feiping Nie, Huimin Chen, Zhanxuan Hu, Rong Wang, and Xuelong Li. Hyperbolic hierarchical representation learning for generalized category discovery. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning. 9 2020. URL `http://arxiv.org/abs/2006.07733`.

William W Hager. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

Katherine A Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304, 2005.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

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*, pages 6418–6428, 2020.

Sungyeon Kim, Boseung Jeong, and Suha Kwak. HIER: Metric Learning Beyond Class Labels via Hierarchical Regularization. 4 2023. URL http://arxiv.org/abs/2212.14258.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Timo Klein, Thomas Lang, Andrii Shkabrii, Alexander Sturm, Kevin Sidak, Lukas Miklautz, Claudia Plant, Yllka Velaj, and Sebastian Tschiatschek. Understanding and improving hyperbolic deep reinforcement learning. *arXiv preprint arXiv:2512.14202*, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Thomas Lang, Kevin Sidak, Claudia Plant, and Yllka Velaj. Hyperbolic math: A pytorch library for hyperbolic deep learning, 2026. URL https://doi.org/10.5281/zenodo.19236209.

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Fangfei Lin, Bing Bai, Yiwen Guo, Hao Chen, Yazhou Ren, and Zenglin Xu. Mhcn: A hyperbolic neural network model for multi-view hierarchical clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16525–16535, 2023.

Jiří Matoušek. On embedding trees into uniformly convex banach spaces. *Israel Journal of Mathematics*, 114(1):221–237, 1999.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Nicholas Monath, Manzil Zaheer, Daniel Silva, Andrew McCallum, and Amr Ahmed. Gradient-based hierarchical clustering using continuous representations of trees in hyperbolic space. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 714–722, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330997. URL https://doi.org/10.1145/3292500.3330997.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pages 3779–3788. PMLR, 2018.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. 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 Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Qi Qian. Stable cluster discrimination for deep clustering. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 16599–16608. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01526.

John G Ratcliffe. *Foundations of hyperbolic manifolds*. Springer, 2006.

Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. Representation tradeoffs for hyperbolic embeddings. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4460–4469. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/sala18a.html.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International symposium on graph drawing*, pages 355–366. Springer, 2011.

Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. In *International Conference on Learning Representations (ICLR)*, 2021.

Samuel Stevens, Jiaman Wu, Matthew J Thompson, Elizabeth G Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila M Dahdul, Charles Stewart, Tanya Berger-Wolf, et al. Bioclip: A vision foundation model for the tree of life. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19412–19424. IEEE, 2024.

Robert van der Klis, Ricardo Chávez Torres, Max van Spengler, Yuhui Ding, Thomas Hofmann, and Pascal Mettes. Fast and geometrically grounded lorentz neural networks. *arXiv preprint arXiv:2601.21529*, 2026.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist Species Classification and Detection Dataset. 4 2018. URL http://arxiv.org/abs/1707.06642.

Dingkang Wang and Yusu Wang. An improved cost function for hierarchical cluster trees. *arXiv preprint arXiv:1812.02715*, 2018.

Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 478–487. JMLR.org, 2016. URL https://dl.acm.org/doi/10.5555/3045390.3045442.

Jiexi Yan, Lei Luo, Cheng Deng, and Heng Huang. Unsupervised hyperbolic metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12465–12474, 2021.

Menglin Yang, Min Zhou, Rex Ying, Yankai Chen, and Irwin King. Hyperbolic representation learning: Revisiting and advancing. In *International conference on machine learning*, pages 39639–39659. PMLR, 2023.

Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao Li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Comput. Surv.*, 57(3):69:1–69:38, 2025. doi: 10.1145/3689036. URL https://doi.org/10.1145/3689036.

A Technical appendices and supplementary material

Table 4 summarizes the contents of our Appendix:

Appendix Section	Content
Appendix A.1	Parameter Summary Table
Appendix A.2	Hyperbolic Space Operations
Appendix A.3	Loss Function & Sampling
Appendix A.4	Theorems & Proofs
Appendix A.5	Datasets & Evaluation Metrics
Appendix A.6	Experiment Details
Appendix A.7	Ablation Studies

Table 4: Structure of our Appendix.

A.1 Parameter Summary Table

Table 5 summarizes the notation used throughout the paper.

Symbol	Description
General	
d	Latent / clustering space dimension
n	Dataset size
B	Batch size
$\boldsymbol{Z}$	Datapoints embedded into the hyperbolic latent space
Hyperbolic geometry	
$\mathcal{M}_c^d$	General hyperbolic space (Poincaré Ball $\mathbb{P}_c^d$ or Hyperboloid $\mathbb{H}_c^d$)
$\bar{\mathbf{0}}$	Hyperbolic space origin
$\mathcal{T}_{\boldsymbol{x}}\mathcal{M}_c^d$	Tangent space at $x \in \mathcal{M}_c^d$
$\exp_{\boldsymbol{x}}$	Exponential map at $x \in \mathcal{M}_c^d$
$d_{\mathcal{M}_c^d}(\boldsymbol{x}, \boldsymbol{y})$	Geodesic distance: length of the geodesic segment connecting $\boldsymbol{x}$ and $\boldsymbol{y}$
$\gamma_{\boldsymbol{x},\boldsymbol{y}}$	Geodesic segment connecting $\boldsymbol{x}$ and $\boldsymbol{y}$
$\Gamma_{\boldsymbol{x},\boldsymbol{y}}$	Geodesic line connecting $\boldsymbol{x}$ and $\boldsymbol{y}$ (extended indefinitely)
$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\boldsymbol{z})$	Geodesic orthogonal projection of $\boldsymbol{z}$ onto $\Gamma_{\boldsymbol{x},\boldsymbol{y}}$
$\mathrm{LCA}(\cdot,\cdot)$	Hyperbolic Lowest Common Ancestor
Loss function	
$\mathcal{L}_{\mathrm{C}}$	Contrastive InfoNCE loss with weight λ_{C}
$\mathcal{L}_{\mathrm{H}}$	Hierarchical (hyperbolic) loss with weight λ_{H}
w_{ij}	Pairwise similarity between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$, bounded by $w_{\min}$ and $w_{\max}$
N_t	Unnormalized cost of a single triplet $t = (i, j, k)$
$N_t^{\min}$	Lower cost bound for triplet t
$N_t^{\max}$	Upper cost bound for triplet t
σ_τ	Temperature-scaled softmax function
Sample complexity (Theorem 4.5)	
ε	Maximum absolute deviation from the true hierarchical loss
δ	Failure probability for the loss approximation (confidence $1 - \delta$)
m	Minimum triplet sample size

Table 5: Notation used throughout the paper.

A.2 Hyperbolic Space Operations

A.2.1 Möbius Addition

In the Poincaré Ball, *Möbius addition* $\boldsymbol{x} \oplus \boldsymbol{y}$ serves as the hyperbolic analogue to Euclidean vector translation $\boldsymbol{x} + \boldsymbol{y}$. Since the Poincaré Ball is a bounded Riemannian manifold, standard Euclidean addition is not closed. To preserve the geometric structure and ensure the resulting point remains strictly within the manifold, Möbius addition is defined as:

$$\boldsymbol{x} \oplus \boldsymbol{y} = \frac{(1 + 2c\langle \boldsymbol{x}, \boldsymbol{y} \rangle + c\|\boldsymbol{y}\|^2)\boldsymbol{x} + (1 - c\|\boldsymbol{x}\|^2)\boldsymbol{y}}{1 + 2c\langle \boldsymbol{x}, \boldsymbol{y} \rangle + c^2\|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2} \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{P}_c^d.$$

A.2.2 Distance

For any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{M}_c^d$, the *geodesic distance* $d_{\mathcal{M}_c^d}(\boldsymbol{x}, \boldsymbol{y})$ represents the length of the shortest path connecting them along the hyperbolic manifold.

Manifold	**Geodesic Distance**
Poincaré Ball	$d_{\mathcal{M}_c^d}(\boldsymbol{x}, \boldsymbol{y}) = \frac{2}{\sqrt{c}} \text{artanh}\left(\sqrt{c}\,\|-\boldsymbol{x} \oplus \boldsymbol{y}\|\right)$
Hyperboloid	$d_{\mathcal{M}_c^d}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\sqrt{c}} \text{arcosh}\left(-c\,\langle \boldsymbol{x}, \boldsymbol{y} \rangle_L\right)$

Table 6: Geodesic distance formulas for the Poincaré Ball and Hyperboloid.

A.2.3 Parallel Transport

Parallel transport is a linear isometry $PT_{\boldsymbol{x} \to \boldsymbol{y}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{M}_c^d \to \mathcal{T}_{\boldsymbol{y}}\mathcal{M}_c^d$ mapping tangent vectors $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathcal{M}_c^d$ along geodesics to other tangent spaces $\mathcal{T}_{\boldsymbol{y}}\mathcal{M}_c^d$. Parallel transport allows us to move the local geometry of a manifold along geodesics and thus defines a way to connect the geometries of nearby points.

Manifold	**Parallel Transport**
Poincaré Ball	$PT_{\boldsymbol{x} \to \boldsymbol{y}}(\boldsymbol{v}) = \frac{\lambda_{\boldsymbol{x}}^c}{\lambda_{\boldsymbol{y}}^c}\left(-(\boldsymbol{y} \oplus -\boldsymbol{x}) \oplus (\boldsymbol{y} \oplus (-\boldsymbol{x} \oplus \boldsymbol{v}))\right)$
Hyperboloid	$PT_{\boldsymbol{x} \to \boldsymbol{y}}(\boldsymbol{v}) = \boldsymbol{v} + \frac{\langle \boldsymbol{v}, \boldsymbol{y} \rangle_L}{\frac{1}{c} - \langle \boldsymbol{x}, \boldsymbol{y} \rangle_L}(\boldsymbol{x} + \boldsymbol{y})$

Table 7: Parallel transport formulas for the Poincaré Ball and Hyperboloid.

A.2.4 Exponential Map

The *exponential map* $\exp_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathcal{M}_c^d \to \mathcal{M}_c^d$ at a point $\boldsymbol{x} \in \mathcal{M}_c^d$ maps tangent vectors $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathcal{M}_c^d$ to the manifold such that the curve $t \in [0, 1] \mapsto \exp_{\boldsymbol{x}}(t\boldsymbol{v})$ is a geodesic joining $\boldsymbol{x} = \exp_{\boldsymbol{x}}(\boldsymbol{0})$ and $\exp_{\boldsymbol{x}}(\boldsymbol{v})$. Since geodesics are only defined locally, the exponential map, which relies on their existence, is also only defined locally. However, in geodesically complete manifolds such as the Poincaré Ball or the Hyperboloid, these local definitions can be extended indefinitely, allowing the exponential map to be defined on the entire tangent space $\mathcal{T}_{\boldsymbol{x}}\mathcal{M}_c^d$.

Manifold	**Exponential Map**
Poincaré Ball	$\exp_{\boldsymbol{x}}(\boldsymbol{v}) = \boldsymbol{x} \oplus \left(\frac{\tanh\left(\frac{\sqrt{c}\,\lambda_{\boldsymbol{x}}^{c} \|\boldsymbol{v}\|}{2} \right)}{\sqrt{c}\,\|\boldsymbol{v}\|} \right) \boldsymbol{v}$
Hyperboloid	$\exp_{\boldsymbol{x}}(\boldsymbol{v}) = \cosh\left(\sqrt{c}\,\|\boldsymbol{v}\|_L\right) \boldsymbol{x} + \sinh\left(\sqrt{c}\,\|\boldsymbol{v}\|_L\right) \frac{\boldsymbol{v}}{\sqrt{c}\,\|\boldsymbol{v}\|_L}$

Table 8: Exponential map formulas for the Poincaré Ball and Hyperboloid.

A.2.5 Feature Scaling

Feature scaling is a method to mitigate optimization instabilities in hyperbolic representation learning. It prevents the norms of Euclidean embeddings from growing excessively large, thus effectively bounding the gradients of the hyperbolic layers while preserving the model's maximal representational capacity [Klein et al., 2025]. To ensure gradient stability, the Euclidean embedding norms are bounded via a learnable scalar $\xi_\theta \in \mathbb{R}$, which scales Euclidean embeddings $\boldsymbol{x} \in \mathbb{R}^d$ as follows:

$$\boldsymbol{x}^{\text{scaled}} = \sigma(\xi_\theta)\, \frac{\operatorname{atanh}(\alpha)}{\sqrt{c}}\, \frac{f(\operatorname{RMS}(\boldsymbol{x}))}{\sqrt{d}},$$

where $\sigma(.)$ is the sigmoid function, $\alpha \in (0, 1)$, and $\operatorname{RMS}(\boldsymbol{x}) = \boldsymbol{x}/\mu(\boldsymbol{x})$ with $\mu(\boldsymbol{x}) = \sqrt{\varepsilon + \frac{1}{d}\|\boldsymbol{x}\|^2}$. In our experiments, we choose $\alpha = 0.95$ and f to be the identity function, as it satisfies the desired properties of being 1-Lipschitz continuous with a fixed point at 0.

A.2.6 Hyperbolic Linear Layers

In Euclidean space, a standard affine transformation of the input $\boldsymbol{x} \in \mathbb{R}^d$ along the k-th output coordinate takes the form

$$v_k(\boldsymbol{x}) = \langle \boldsymbol{a}_k, \boldsymbol{x} \rangle - b_k = \langle \boldsymbol{a}_k, -\boldsymbol{p}_k + \boldsymbol{x} \rangle, \qquad b_k = \langle \boldsymbol{a}_k, \boldsymbol{p}_k \rangle$$

which can be interpreted as a signed distance from $\boldsymbol{x}$ to the hyperplane with normal $\boldsymbol{a}_k \in \mathbb{R}^d$ and offset $\boldsymbol{p}_k \in \mathbb{R}^d$:

$$v_k(\boldsymbol{x}) = \langle \boldsymbol{a}_k, -\boldsymbol{p}_k + \boldsymbol{x} \rangle = \|\boldsymbol{a}_k\| \operatorname{sign}(\langle \boldsymbol{a}_k, -\boldsymbol{p}_k + \boldsymbol{x} \rangle)\, d(\boldsymbol{x}, H_{\boldsymbol{a}_k, \boldsymbol{p}_k})$$

This signed-distance interpretation admits a natural generalization to hyperbolic geometry. Ganea et al. [2018] introduced this hyperbolic counterpart using the algebraic structure of the Poincaré Ball:

$$\tilde{v}_k(\boldsymbol{x}) = \|\boldsymbol{a}_k\|_{\mathcal{T}_{\boldsymbol{p}_k}\mathcal{M}_c^d} \operatorname{sign}\left(\langle \boldsymbol{a}_k,\, -\boldsymbol{p}_k \oplus \boldsymbol{x} \rangle\right) d_{\mathcal{M}_c^d}(\boldsymbol{x},\, \tilde{H}_{\boldsymbol{a}_k, \boldsymbol{p}_k}),$$

where $\oplus$ denotes the Möbius addition and $\tilde{H}_{\boldsymbol{a}_k, \boldsymbol{p}_k} = \{\boldsymbol{z} \in \mathbb{P}_c^d : \langle \boldsymbol{a}_k, -\boldsymbol{p}_k \oplus \boldsymbol{z} \rangle = 0\}$ is the corresponding geodesic hyperplane. While $\tilde{v}_k(\boldsymbol{x})$ was originally formulated using Poincaré Ball specific operations, the underlying geometric principle – computing the distance to a geodesic hyperplane defined by a normal vector $\boldsymbol{a}_k$ and offset $\boldsymbol{p}_k$ – applies to any hyperbolic manifold $\mathcal{M}_c^d$.

The hyperbolic linear layer is constructed by computing the signed geodesic distances $\tilde{v}_k(\boldsymbol{x})$ from the input to a set of learned geodesic hyperplanes and mapping the result back to the manifold as the unique point $\boldsymbol{y} \in \mathcal{M}_c^K$ whose distances to the coordinate hyperplanes through the manifold's origin match $\tilde{v}_k(\boldsymbol{x})$. For both hyperbolic space models, we learn tangent vectors $\{\boldsymbol{z}_k \in \mathcal{T}_{\bar{\mathbf{0}}}\mathcal{M}_c^d\}_{k=1}^K$ at the origin $\bar{\mathbf{0}}$ and scalar offsets $\{r_k \in \mathbb{R}\}_{k=1}^K$, from which the hyperplane normal $\boldsymbol{a}_k \in \mathcal{T}_{\boldsymbol{p}_k}\mathcal{M}_c^d$ and offset $\boldsymbol{p}_k \in \mathcal{M}_c^d$ are recovered via the exponential map (Appendix A.2.4) and parallel transport (Appendix A.2.3):

$$\boldsymbol{a}_k = PT_{\bar{\mathbf{0}} \to \boldsymbol{p}_k}(\boldsymbol{z}_k), \qquad \boldsymbol{p}_k = \exp_{\bar{\mathbf{0}}}\left(r_k \frac{\boldsymbol{z}_k}{\|\boldsymbol{z}_k\|} \right).$$

Poincaré Ball [Shimizu et al., 2021]. The output $\boldsymbol{y} \in \mathbb{P}_c^K$ for input $\boldsymbol{x} \in \mathbb{P}_c^d$ is

$$\boldsymbol{y} = \frac{\boldsymbol{w}}{1 + \sqrt{1 + c\|\boldsymbol{w}\|^2}}, \qquad w_k = \frac{1}{\sqrt{c}} \sinh\big(\sqrt{c}\, \tilde{v}_k(\boldsymbol{x})\big)\,,$$

$$\tilde{v}_k(\boldsymbol{x}) = \frac{2\,\|\boldsymbol{z}_k\|}{\sqrt{c}} \sinh^{-1}\left((1 - \lambda_{\boldsymbol{x}}^c)\, \sinh(2\sqrt{c}\, r_k) + \sqrt{c}\, \lambda_{\boldsymbol{x}}^c\, \cosh(2\sqrt{c}\, r_k)\, \left\langle \frac{\boldsymbol{z}_k}{\|\boldsymbol{z}_k\|}, \boldsymbol{x} \right\rangle\right),$$

where $\lambda_{\boldsymbol{x}}^c = \frac{2}{1 - c\|\boldsymbol{x}\|^2}$ is the conformal factor at $\boldsymbol{x}$.

Hyperboloid [van der Klis et al., 2026]. The output $\boldsymbol{y} \in \mathbb{H}_c^K$ for input $\boldsymbol{x} \in \mathbb{H}_c^d$ is

$$y_0 = \sqrt{\|\boldsymbol{y}_{\mathbf{1:K}}\|^2 + \frac{1}{c}}, \qquad \boldsymbol{y}_{\mathbf{1:K}} = \frac{\sinh\left(\sqrt{c}\, \tilde{\boldsymbol{v}}(\boldsymbol{x})\right)}{\sqrt{c}},$$

$$\tilde{v}_k(\boldsymbol{x}) = \frac{\|\boldsymbol{z}_k\|}{\sqrt{c}}\, \sinh^{-1}\left(-\sqrt{c}\, x_0\, \sinh(\sqrt{c}\, r_k) + \sqrt{c} \cosh(\sqrt{c}\, r_k) \left\langle \frac{\boldsymbol{z}_k}{\|\boldsymbol{z}_k\|}, \boldsymbol{x}_{\mathbf{1:d}} \right\rangle\right),$$

where $\boldsymbol{z_k} \in \mathbb{R}^d$ denotes the spatial coordinates of the tangent vector $(0, \boldsymbol{z_k})$, and y_0 and $\boldsymbol{y}_{\mathbf{1:K}}$ the time and space components of the output, respectively. Note that this formulation is a direct adaptation of the Poincaré Ball linear layer of Shimizu et al. [2021] to the Hyperboloid and does not include the activation functions or rescaling of van der Klis et al. [2026].

All operations introduced in this section are implemented using the `hyperbolic-math` library of Lang et al. [2026] (Zenodo DOI 10.5281/zenodo.19236209), which we use as an external dependency.

A.3 Loss Function & Sampling

A.3.1 Contrastive Loss Formulation

The contrastive loss is the InfoNCE objective [Oord et al., 2018]. For a batch of B images, each is augmented twice to produce $2B$ views, embedded into the latent space, and the loss is computed as

$$\mathcal{L}_{\mathrm{C}} = \frac{1}{2B} \sum_{k=1}^{2B} -\log\left(\frac{\exp(\mathrm{sim}(\boldsymbol{z}_k, \tilde{\boldsymbol{z}}_k)/\tau_{\text{contr}})}{\sum_{l=1}^{2B} \mathbb{1}_{[l \neq k]} \exp(\mathrm{sim}(\boldsymbol{z}_k, \boldsymbol{z}_l)/\tau_{\text{contr}})}\right),$$

where $\boldsymbol{z}_k$ is the embedding of the k-th view, $\tilde{\boldsymbol{z}}_k$ its augmented counterpart, and $\tau_{\text{contr}} > 0$ is the temperature parameter controlling the concentration of the distribution over negatives. The similarity function used depends on the training phase. During pretraining (Appendix A.6.3) it is the cosine similarity between the Euclidean projection-head outputs and during the joint training (Appendix A.6.4) it is the negative geodesic distance in the manifold,

$$\mathrm{sim}(\cdot, \cdot) = -d_{\mathcal{M}_c^d}(\cdot, \cdot).$$

The augmentation family and per-dataset batch sizes are listed in Appendix A.6.6.

A.3.2 Computing the Rectified Hyperbolic LCA

Let $\boldsymbol{p}$ be the projection of $\bar{\mathbf{0}}$ onto the full geodesic passing through $\boldsymbol{x}$ and $\boldsymbol{y}$, given by Propositions 4.3 and 4.4. If $\boldsymbol{p}$ lies on the connecting segment, it is the rectified LCA. Otherwise, the LCA is the endpoint closer to $\bar{\mathbf{0}}$.

We distinguish these cases by testing segment membership using Euclidean angles at the ambient coordinate origin O, including the time coordinate on the Hyperboloid. Since the direction from O varies monotonically along the geodesic, $\boldsymbol{p}$ lies on the segment if both $\angle_{\boldsymbol{x}O\boldsymbol{p}}$ and $\angle_{\boldsymbol{y}O\boldsymbol{p}}$ are at most $\angle_{\boldsymbol{x}O\boldsymbol{y}}$, which we check by comparing cosines. Radial geodesics in the Poincaré Ball are handled directly: opposite rays give LCA depth zero, and points on the same ray give the closer endpoint. In either hyperbolic manifold, this test requires only a constant number of Euclidean inner products.

A.3.3 Importance-Rejection Sampling

The importance sampling approach draws triplets from each batch according to the target distribution $P(t) \propto N_t^{\max} - N_t^{\min}$ and computes the estimator:

$$\mathcal{L}_{\mathrm{H}}^{m}(\boldsymbol{Z}) = \frac{1}{m} \sum_{l=1}^{m} \frac{N_{t_l} - N_{t_l}^{\min}}{N_{t_l}^{\max} - N_{t_l}^{\min}}, \qquad t_l \sim P.$$

Sampling directly from $P(t)$ would require evaluating the normalizing constant $\mathcal{Z} = \sum_{t'} (N_{t'}^{\max} - N_{t'}^{\min})$, a computation of cubic complexity. We avoid this via rejection sampling with the uniform proposal distribution Q:

$$P(t) = \frac{N_t^{\max} - N_t^{\min}}{\mathcal{Z}}, \qquad Q(t) = \frac{1}{|T|},$$

where $|T|$ denotes the number of distinct unordered triplets of the current batch. Triplets with zero spread have zero acceptance probability and thus do not contribute to the loss. To ensure a valid acceptance probability (i.e. $A(t) \leq 1$), we require a bounding constant M such that $P(t) \leq MQ(t)$ for all t. Since any two of the three pairwise sums $\{w_{ij}+w_{ik}, w_{ij}+w_{jk}, w_{ik}+w_{jk}\}$ share a common term, $N_t^{\max} - N_t^{\min}$ is bounded by the difference of the two batch similarity extrema $w_{\max} - w_{\min}$. Setting $M = \frac{|T|(w_{\max} - w_{\min})}{\mathcal{Z}}$ then yields the acceptance probability as the ratio of the target density to the scaled proposal density:

$$A(t) = \frac{P(t)}{MQ(t)} = \frac{\frac{N_t^{\max} - N_t^{\min}}{\mathcal{Z}}}{\frac{|T|(w_{\max} - w_{\min})}{\mathcal{Z}} \cdot \frac{1}{|T|}} = \frac{N_t^{\max} - N_t^{\min}}{w_{\max} - w_{\min}},$$

in which $\mathcal{Z}$ cancels entirely. In practice, we simply draw triplets uniformly at random with replacement and accept them with probability $A(t)$, sidestepping the intractable normalization constant. Since candidates and normalizers relate to the same batch, the bounding condition holds exactly and the accepted triplets are distributed in proportion to $N_t^{\max} - N_t^{\min}$, as required.

Notably, our triplet selection is gradient-free: candidates are accepted or rejected from detached similarity lookups that reuse the geodesic-distance matrix already computed for $\mathcal{L}_{\mathrm{C}}$, requiring only a single elementwise transformation to extract the similarities w_{ij} alongside the normalizers $w_{\min}$ and $w_{\max}$. The LCA-depth and softmax terms of $\mathcal{L}_{\mathrm{H}}$ are then assembled only for the m accepted triplets. The backward pass, which dominates both time and memory, therefore scales solely with the accepted budget m and not with the $\approx m/A$ candidates drawn. Under the observed acceptance rates ($A = 0.02$ on CIFAR-100, 0.05 on TinyImageNet, 0.12 on ImageNet-1k, 0.10 on iNaturalist-2018), satisfying the budget of $m \approx 27{,}000$ requires 0.2–1.4 M candidates per step. Table 9 reports the measured per-step cost at the two endpoints of this range: CIFAR-100 (lowest acceptance rate; highest candidate count) and ImageNet-1k (highest acceptance rate; lowest candidate count). TinyImageNet and iNaturalist-2018 fall between these extremes. Even in the worst case, the candidate count remains orders of magnitude below exhaustive enumeration, and the sampler accounts for only a small fraction of total step time.

Dataset	A	**Candidates / step**	**Global triplets avoided**	**Wall-clock / step**
CIFAR-100	0.02	≈ 1.35 M	$\approx 10^{13}$	12.2 ms (4.4% of step)
ImageNet-1k	0.12	≈ 0.23 M	$\approx 10^{17}$	1.3 ms (0.6% of step)

Table 9: Per-step cost of the importance-rejection sampler at the fixed budget of $m \approx 27{,}000$ for $(\varepsilon, \delta) = (0.01, 0.01)$ according to Theorem 4.5 at the two endpoints of the observed acceptance rates.

A.3.4 Batch-Level Normalizers $w_{\min}$ and $w_{\max}$

With candidates, similarities, and normalizers all drawn from the current batch (Appendix A.3.3), a natural concern is sampling bias since the batch range is generally narrower than the global one (e.g. globally extreme pairs such as maximally dissimilar taxa on iNaturalist-2018 rarely co-occur

in small batches). So one might expect the accepted triplets to over-represent whichever part of the similarity spectrum the batch happens to cover. However, within a fixed batch, using its similarity range does not bias the accepted-triplet distribution: $w_{\max} - w_{\min}$ is a single scalar dividing the spread $N_t^{\max} - N_t^{\min}$ of every candidate, so it cancels from all relative acceptance probabilities

$$A(t)/A(t') = (N_t^{\max} - N_t^{\min})/(N_{t'}^{\max} - N_{t'}^{\min}).$$

A narrower batch-level normalizer thus raises the acceptance rate uniformly without altering the target triplet distribution within that batch. Since the number of accepted triplets is fixed at the budget $m \approx 27{,}000$ of Theorem 4.5, the acceptance rate determines only how many gradient-free candidates are drawn per step (Table 9), and the (ε, δ) guarantee is unaffected.

A.3.5 Regularity of $N_t^{\min}$ and $N_t^{\max}$

$N_t^{\min}$ and $N_t^{\max}$ are defined as the minimum and maximum of the three pairwise sums $\{w_{ij} + w_{ik},\, w_{ij} + w_{jk},\, w_{ik} + w_{jk}\}$. Since each of these sums is smooth with respect to the embeddings wherever the geodesic distance is smooth, their extrema are piecewise smooth. Consequently, they admit subgradients everywhere, are differentiable except on the measure-zero set where two sums coincide, and fall within the regularity class of ReLU and max-pooling. Automatic differentiation handles them identically, selecting a valid subgradient at ties. Away from ties, automatic differentiation computes the exact gradient of each evaluated triplet. Theorem 4.5 is unaffected, as it bounds the estimator's concentration and relies only on the pointwise bounds $N_t^{\min} \le N_t \le N_t^{\max}$, not on differentiability.

Softening the extrema with temperature-weighted soft-$\min$ and soft-$\max$ operators would restore full differentiability, but break precisely these pointwise bounds that our theory rests on. Specifically, N_t and the soft operators are convex combinations of the same three pairwise sums, yet weighted by different quantities: the LCA depths for N_t, and the pairwise similarities for the soft operators. Since nothing forces these weightings to align, N_t can concentrate near the hard minimum and fall below the soft-$\min$, rendering the per-triplet ratio negative and invalidating the variance bound used in Theorem 4.5. Ultimately, the hard formulation is exact, bounded, and differentiable almost everywhere, making it sufficient for all practical purposes.

A.4 Theorems & Proofs

This Appendix section is split into three subsections: the proof of the geodesic orthogonal projections (Proposition 4.3 & 4.4), the proof of the minimum sample complexity (Theorem 4.5), and proof of the fixed-similarity equivalence (Proposition 4.1).

A.4.1 Geodesic Orthogonal Projections

Mathematical Preliminaries. Before deriving the explicit formulas for geodesic orthogonal projections in the Poincaré Ball and the Hyperboloid, we briefly recap the form of geodesics in each hyperbolic space:

- **Poincaré Ball.** In the Poincaré Ball, geodesics are arcs of circles that intersect the boundary sphere $\partial\mathbb{P}_c^d$ orthogonally. In the special case where these circles pass through the origin, the geodesics are straight lines through the origin, which occurs when the points are collinear with the origin.
- **Hyperboloid.** In the Hyperboloid, geodesics are intersections of $\mathbb{H}_c^d$ with 2-dimensional linear subspaces of the ambient Minkowski space $\mathbb{R}^{1,d}$.

Lemma A.1 (Sherman-Morrison)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ be an invertible square matrix and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d$ be column vectors. Then $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\top$ is invertible if and only if $1 + \boldsymbol{v}^\top \boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$. In this case, its inverse is given by:*

$$(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\top)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^\top\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^\top\boldsymbol{A}^{-1}\boldsymbol{u}}.$$

Proof. For the proof, we refer the reader to the original paper by Hager [1989]. □

Geodesic Orthogonal Projections in the Poincaré Ball

Proposition 4.3. *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{P}_c^d$ be non-collinear. Then*

$$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) = \left(1 - \frac{\sqrt{\|\boldsymbol{z}\|^2 - \frac{1}{c}}}{\|\boldsymbol{z}\|}\right)\boldsymbol{z},$$

$$\boldsymbol{z} = \frac{\left[(1+c\|\boldsymbol{y}\|^2)\langle \boldsymbol{x},\boldsymbol{y}\rangle - (1+c\|\boldsymbol{x}\|^2)\|\boldsymbol{y}\|^2\right]\boldsymbol{x} + \left[(1+c\|\boldsymbol{x}\|^2)\langle \boldsymbol{x},\boldsymbol{y}\rangle - (1+c\|\boldsymbol{y}\|^2)\|\boldsymbol{x}\|^2\right]\boldsymbol{y}}{2c(\langle \boldsymbol{x},\boldsymbol{y}\rangle^2 - \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2)}.$$

Proof. By assumption, $\boldsymbol{x}$ and $\boldsymbol{y}$ are non-collinear, i.e. the geodesic $\Gamma_{\boldsymbol{x},\boldsymbol{y}}$ is an arc of a circle that meets the boundary of $\mathbb{P}_c^d$ orthogonally (Appendix A.4.1). Using the circle inversion property with respect to the boundary sphere $\partial\mathbb{P}_c^d = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|^2 = 1/c\}$, we can compute the inverses of $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\boldsymbol{x}^{-1} = \frac{\boldsymbol{x}}{c\|\boldsymbol{x}\|^2}, \qquad \boldsymbol{y}^{-1} = \frac{\boldsymbol{y}}{c\|\boldsymbol{y}\|^2}.$$

These points satisfy $\|\boldsymbol{x}\| \cdot \|\boldsymbol{x}^{-1}\| = 1/c$ and $\|\boldsymbol{y}\| \cdot \|\boldsymbol{y}^{-1}\| = 1/c$.

1. **Finding conditions on the circle center.** By construction, the points $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}^{-1}, \boldsymbol{y}^{-1}$ all lie on the same supporting Euclidean circle $\mathcal{C}_{\boldsymbol{x},\boldsymbol{y}}$, whose arc inside the ball is $\Gamma_{\boldsymbol{x},\boldsymbol{y}}$. Thus, we can construct two perpendicular-bisecting hyperplanes to derive conditions the center $\boldsymbol{z}$ has to satisfy.

   - First perpendicular bisector condition:

   $$\left\langle \boldsymbol{z} - \frac{\boldsymbol{x} + \boldsymbol{x}^{-1}}{2}, \boldsymbol{x} - \boldsymbol{x}^{-1} \right\rangle = 0 \iff \langle \boldsymbol{z}, \boldsymbol{x}\rangle = \frac{1 + c\|\boldsymbol{x}\|^2}{2c}. \tag{1}$$

   - Second perpendicular bisector condition:

   $$\left\langle \boldsymbol{z} - \frac{\boldsymbol{y} + \boldsymbol{y}^{-1}}{2}, \boldsymbol{y} - \boldsymbol{y}^{-1} \right\rangle = 0 \iff \langle \boldsymbol{z}, \boldsymbol{y}\rangle = \frac{1 + c\|\boldsymbol{y}\|^2}{2c}. \tag{2}$$

   - Span condition:

   $$\boldsymbol{z} = \alpha\boldsymbol{x} + \beta\boldsymbol{y}. \tag{3}$$

2. **Solving for the coefficients.** Combining equations (1) and (3), as well as (2) and (3):

   $$\langle \alpha\boldsymbol{x} + \beta\boldsymbol{y}, \boldsymbol{x}\rangle = \alpha\|\boldsymbol{x}\|^2 + \beta\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \frac{1 + c\|\boldsymbol{x}\|^2}{2c}, \tag{4}$$

   $$\langle \alpha\boldsymbol{x} + \beta\boldsymbol{y}, \boldsymbol{y}\rangle = \alpha\langle \boldsymbol{x}, \boldsymbol{y}\rangle + \beta\|\boldsymbol{y}\|^2 = \frac{1 + c\|\boldsymbol{y}\|^2}{2c}. \tag{5}$$

   Solving the first combined equation (4) for α:

   $$\alpha = \frac{1 + c\|\boldsymbol{x}\|^2}{2c\|\boldsymbol{x}\|^2} - \frac{\beta\langle \boldsymbol{x}, \boldsymbol{y}\rangle}{\|\boldsymbol{x}\|^2}$$

   Substituting α into the second combined equation (5) determines β:

   $$\beta = \frac{(1+c\|\boldsymbol{x}\|^2)\langle \boldsymbol{x},\boldsymbol{y}\rangle - (1+c\|\boldsymbol{y}\|^2)\|\boldsymbol{x}\|^2}{2c(\langle \boldsymbol{x},\boldsymbol{y}\rangle^2 - \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2)} \tag{6}$$

   Finally, we solve for α:

   $$\alpha = \frac{(1+c\|\boldsymbol{y}\|^2)\langle \boldsymbol{x},\boldsymbol{y}\rangle - (1+c\|\boldsymbol{x}\|^2)\|\boldsymbol{y}\|^2}{2\,c\,(\langle \boldsymbol{x},\boldsymbol{y}\rangle^2 - \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2)}. \tag{7}$$

3. **Center of the circle.** The center $\boldsymbol{z}$ of the circle $\mathcal{C}_{\boldsymbol{x},\boldsymbol{y}}$ is then fully determined through its coefficients α (7) and β (6):

$$\boldsymbol{z} = \frac{\left[(1 + c\|\boldsymbol{y}\|^2)\langle \boldsymbol{x}, \boldsymbol{y}\rangle - (1 + c\|\boldsymbol{x}\|^2)\|\boldsymbol{y}\|^2\right]\boldsymbol{x} + \left[(1 + c\|\boldsymbol{x}\|^2)\langle \boldsymbol{x}, \boldsymbol{y}\rangle - (1 + c\|\boldsymbol{y}\|^2)\|\boldsymbol{x}\|^2\right]\boldsymbol{y}}{2c(\langle \boldsymbol{x}, \boldsymbol{y}\rangle^2 - \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2)}$$

4. **Geodesic projection.** Since the circle $\mathcal{C}_{\boldsymbol{x},\boldsymbol{y}}$ meets the boundary orthogonally, we have:

$$\frac{1}{c} = \|\boldsymbol{z}\|^2 - r^2 \quad \Longleftrightarrow \quad r = \sqrt{\|\boldsymbol{z}\|^2 - \frac{1}{c}} \tag{8}$$

where r is the radius of the supporting circle $\mathcal{C}_{\boldsymbol{x},\boldsymbol{y}}$. Moreover, we have that the geodesic orthogonal projection $\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}})$ lies on the line from the origin to $\boldsymbol{z}$, and is the point on $\Gamma_{\boldsymbol{x},\boldsymbol{y}}$ closest to the origin. This gives us:

$$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) = (\|\boldsymbol{z}\| - r)\,\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|} \tag{9}$$

Combining equations (8) and (9) yields the formula for the geodesic orthogonal projection:

$$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) = (\|\boldsymbol{z}\| - r)\,\frac{\boldsymbol{z}}{\|\boldsymbol{z}\|} = \left(1 - \frac{\sqrt{\|\boldsymbol{z}\|^2 - \frac{1}{c}}}{\|\boldsymbol{z}\|}\right)\boldsymbol{z}.$$

□

Geodesic Orthogonal Projections in the Hyperboloid

Proposition 4.4. *Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}^d_c$ with $x \neq y$. Then*

$$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) = \frac{\alpha\boldsymbol{x} + \beta\boldsymbol{y}}{\sqrt{-c\,\langle \alpha\boldsymbol{x} + \beta\boldsymbol{y}, \alpha\boldsymbol{x} + \beta\boldsymbol{y}\rangle_L}},$$

$$\alpha = \frac{\sqrt{c}\,(x_0 + c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L\, y_0)}{1 - (c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L)^2}, \qquad \beta = \frac{\sqrt{c}\,(y_0 + c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L\, x_0)}{1 - (c\langle \boldsymbol{x}, \boldsymbol{y}\rangle_L)^2}.$$

Proof. The geodesic orthogonal projection $\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}})$ is computed in two steps:

1. Compute the Minkowski-orthogonal linear projection $\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}})$ of the origin $\bar{\mathbf{0}}$ onto the linear subspace $\text{span}\{\boldsymbol{x}, \boldsymbol{y}\}$ with respect to the Minkowski inner product $\langle ., . \rangle_L$.
2. Rescale $\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}})$ to be a point on the Hyperboloid:

$$\pi_{\Gamma_{\boldsymbol{x},\boldsymbol{y}}}(\bar{\mathbf{0}}) = \frac{\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}})}{\sqrt{-c\langle \pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}}), \pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}})\rangle_L}}.$$

Let $\boldsymbol{M} = [\boldsymbol{x} \,|\, \boldsymbol{y}] \in \mathbb{R}^{(d+1)\times 2}$ denote the matrix with column-vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{d+1}$, $\bar{\mathbf{0}} \in \mathbb{R}^{d+1}$ the Hyperboloid's origin written as column-vector, $\boldsymbol{B} = \text{diag}[-1, 1, \ldots, 1] \in \mathbb{R}^{(d+1)\times(d+1)}$ the matrix associated with the Minkowski inner product, and $\rho = \langle \boldsymbol{x}, \boldsymbol{y}\rangle_L$. The Minkowski-orthogonal linear projection $\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}})$ with respect to $\langle ., . \rangle_L$ is given by:

$$\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}}) = \boldsymbol{M}(\boldsymbol{M}^\top \boldsymbol{B}\boldsymbol{M})^{-1}\boldsymbol{M}^\top \boldsymbol{B}\,\bar{\mathbf{0}}.$$

Since $\boldsymbol{M}^\top \boldsymbol{B}\boldsymbol{M} \in \mathbb{R}^{2\times 2}$ is a symmetric matrix with main diagonal entries $(\langle \boldsymbol{x}, \boldsymbol{x}\rangle_L, \langle \boldsymbol{y}, \boldsymbol{y}\rangle_L) = (-1/c, -1/c)$ and off-diagonal element ρ we can apply the Sherman-Morrison Lemma A.1 to directly express the inverse. We set $\boldsymbol{A} = (-1/c - \rho)\,\mathbb{I}_{\boldsymbol{2}} \in \mathbb{R}^{2\times 2}$, $\boldsymbol{u} = (\rho, \rho)^\top \in \mathbb{R}^2$, and $\boldsymbol{v} = (1, 1)^\top \in \mathbb{R}^2$. Then $\boldsymbol{A}$ is invertible and $1 + \boldsymbol{v}^\top \boldsymbol{A}^{-1}\boldsymbol{u} = \frac{1-c\rho}{1+c\rho} \neq 0$ since $\boldsymbol{x} \neq \boldsymbol{y}$. Thus:

$$(\boldsymbol{M}^\top \boldsymbol{B}\boldsymbol{M})^{-1} = (\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\top)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^\top \boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^\top \boldsymbol{A}^{-1}\boldsymbol{u}} = \frac{-c}{1 - (c\rho)^2}\begin{bmatrix} 1 & c\rho \\ c\rho & 1 \end{bmatrix}$$

Next, we compute $\boldsymbol{M}^\top \boldsymbol{B}\,\bar{\mathbf{0}} = (-x_0/\sqrt{c}, -y_0/\sqrt{c})^\top \in \mathbb{R}^2$ and put everything together:

$$\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}}) = \boldsymbol{M}(\boldsymbol{M}^\top \boldsymbol{B}\boldsymbol{M})^{-1}\boldsymbol{M}^\top \boldsymbol{B}\,\bar{\mathbf{0}} = [\boldsymbol{x}\,|\,\boldsymbol{y}]\,\frac{-c}{1-(c\rho)^2}\begin{bmatrix}1 & c\rho\\ c\rho & 1\end{bmatrix}\begin{bmatrix}-x_0/\sqrt{c}\\ -y_0/\sqrt{c}\end{bmatrix} =$$

$$= \frac{\sqrt{c}}{1-(c\rho)^2}\left((x_0 + c\rho y_0)\boldsymbol{x} + (c\rho x_0 + y_0)\boldsymbol{y}\right).$$

Re-substituting for ρ yields:

$$\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}}) = \alpha\boldsymbol{x} + \beta\boldsymbol{y},$$

where

$$\alpha = \frac{\sqrt{c}\,(x_0 + c\langle \boldsymbol{x},\boldsymbol{y}\rangle_L\, y_0)}{1-(c\langle \boldsymbol{x},\boldsymbol{y}\rangle_L)^2}, \qquad \beta = \frac{\sqrt{c}\,(y_0 + c\langle \boldsymbol{x},\boldsymbol{y}\rangle_L\, x_0)}{1-(c\langle \boldsymbol{x},\boldsymbol{y}\rangle_L)^2}$$

Finally, rescaling $\pi_{\text{span}\{\boldsymbol{x},\boldsymbol{y}\}}(\bar{\mathbf{0}})$ such that it is a point on the Hyperboloid yields the geodesic orthogonal projection formula. □

A.4.2 Minimum Sample Complexity

Mathematical Preliminaries. To derive a lower bound on the minimum sample size m for our loss function, it is necessary to bound the individual loss components to later apply concentration inequalities. In the following, we use the same notation as in the main paper (Section 4.1).

Proposition A.2. *Let $t = (i, j, k)$ be any triplet of indices,*

$$N_t = w_{ij} + w_{ik} + w_{jk} - \left\langle \begin{pmatrix} w_{ij}\\ w_{ik}\\ w_{jk}\end{pmatrix}, \sigma_\tau \begin{pmatrix} d_{\mathcal{M}_c^d}(\bar{\mathbf{0}}, \mathit{LCA}(\boldsymbol{z}_i, \boldsymbol{z}_j))\\ d_{\mathcal{M}_c^d}(\bar{\mathbf{0}}, \mathit{LCA}(\boldsymbol{z}_i, \boldsymbol{z}_k))\\ d_{\mathcal{M}_c^d}(\bar{\mathbf{0}}, \mathit{LCA}(\boldsymbol{z}_j, \boldsymbol{z}_k))\end{pmatrix}\right\rangle,$$

and

$$\mathcal{L}_H(\boldsymbol{Z}) = \frac{\sum_t (N_t - N_t^{\min})}{\sum_t (N_t^{\max} - N_t^{\min})}.$$

Then the following bounds hold:

$$0 \le N_t - N_t^{\min} \le N_t^{\max} - N_t^{\min}, \qquad 0 \le \mathcal{L}_H(\boldsymbol{Z}) \le 1,$$

where $w_{\min} = \min_{ij}(w_{ij})$ and $w_{\max} = \max_{ij}(w_{ij})$, respectively.

Proof. Recall that

$$N_t = w_{ij} + w_{ik} + w_{jk} - p_{ij}w_{ij} - p_{ik}w_{ik} - p_{jk}w_{jk}$$

where p is a probability distribution from the softmax function. This can be rewritten as:

$$N_t = p_{ij}(w_{ik} + w_{jk}) + p_{ik}(w_{ij} + w_{jk}) + p_{jk}(w_{ij} + w_{ik}).$$

Thus, N_t is a convex combination of the three pairwise sums $\{w_{ij} + w_{ik}, w_{ij} + w_{jk}, w_{ik} + w_{jk}\}$. By definition, $N_t^{\min}$ is the minimum of these three sums, and $N_t^{\max}$ is the maximum. Therefore:

$$N_t^{\min} \le N_t \le N_t^{\max}.$$

Subtracting $N_t^{\min}$ from all sides yields:

$$0 \le N_t - N_t^{\min} \le N_t^{\max} - N_t^{\min}. \tag{10}$$

Summing inequality (10) over all valid triplets t:

$$0 \le \sum_t (N_t - N_t^{\min}) \le \sum_t (N_t^{\max} - N_t^{\min})$$

and dividing all three terms by $\sum_t (N_t^{\max} - N_t^{\min})$ bounds the loss function:

$$0 \le \frac{\sum_t (N_t - N_t^{\min})}{\sum_t (N_t^{\max} - N_t^{\min})} \le \frac{\sum_t (N_t^{\max} - N_t^{\min})}{\sum_t (N_t^{\max} - N_t^{\min})} = 1.$$

□

Lemma A.3 (Bhatia-Davis Inequality)**.** *Let Y be a bounded random variable such that $a \leq Y \leq b$. Then its variance σ^2 satisfies:*

$$\sigma^2 \leq (b - \mathbb{E}[Y])(\mathbb{E}[Y] - a).$$

Proof. For the proof, we refer the reader to the original paper by Bhatia and Davis [2000]. □

Lemma A.4 (Bernstein's Inequality)**.** *Let $Y_1, \ldots, Y_m$ be independent zero-mean random variables with $|Y_l| \leq M$ and Var$(Y_l) \leq \sigma^2$ for all l. Then for any $\tau > 0$,*

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{l=1}^{m} Y_l\right| \geq \tau\right] \leq 2\exp\left(-\frac{m\tau^2}{2\sigma^2 + \frac{2}{3}M\tau}\right).$$

Proof. For the proof, we refer the reader to the original paper by Bernstein [1924]. □

Minimum Sample Complexity - Theorem & Proof

Theorem 4.5 (Minimum Sample Complexity)**.** *For any $\varepsilon, \delta \in (0,1)$, drawing*

$$m \geq \frac{\left(\frac{1}{2} + \frac{2}{3}\varepsilon\right)\ln\left(\frac{2}{\delta}\right)}{\varepsilon^2}$$

triplets from $P(t) \propto N_t^{\max} - N_t^{\min}$ guarantees $\mathbb{P}\left[|\mathcal{L}_H^m(\boldsymbol{Z}) - \mathcal{L}_H(\boldsymbol{Z})| \geq \varepsilon\right] \leq \delta$.

Proof. We prove the bound using Bernstein's inequality (Lemma A.4). For our importance sampling approach, we introduce a surrogate variable that directly captures the normalized loss. Specifically, we define

$$X_t = \frac{N_t - N_t^{\min}}{N_t^{\max} - N_t^{\min}}, \qquad \text{with } t \sim P,$$

where any triplet with $N_t^{\max} = N_t^{\min}$ yields $P(t) = 0$ and is therefore never sampled. The expectation of X_t under the target distribution P is given by:

$$\mathbb{E}_{t\sim P}[X_t] = \sum_{t:P(t)>0} P(t) \cdot \frac{N_t - N_t^{\min}}{N_t^{\max} - N_t^{\min}} = \sum_{t:P(t)>0} \frac{N_t^{\max} - N_t^{\min}}{\sum_{t'}(N_{t'}^{\max} - N_{t'}^{\min})} \cdot \frac{N_t - N_t^{\min}}{N_t^{\max} - N_t^{\min}}$$

$$= \frac{\sum_t (N_t - N_t^{\min})}{\sum_{t'}(N_{t'}^{\max} - N_{t'}^{\min})} = \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z}).$$

This confirms that our empirical estimator $\mathcal{L}_{\mathrm{H}}^m(\boldsymbol{Z}) = \frac{1}{m}\sum_{l=1}^m X_{t_l}$ is an unbiased estimator of the true loss.

From Proposition A.2, we know that $0 \leq N_t - N_t^{\min} \leq N_t^{\max} - N_t^{\min}$, which constrains the surrogate variable to the range $0 \leq X_t \leq 1$. Applying the Bhatia-Davis inequality (Lemma A.3) with bounds $a = 0$ and $b = 1$ provides an upper bound on the variance of X_t:

$$\sigma_{X_t}^2 \leq (1 - \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z}))(\mathcal{L}_{\mathrm{H}}(\boldsymbol{Z}) - 0) = \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z})(1 - \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z})) \leq \frac{1}{4}.$$

To apply Bernstein's inequality (Lemma A.4) to the deviation of the mean, we define the centered random variable $Y_t = X_t - \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z})$. By the linearity of expectation, $\mathbb{E}[Y_t] = 0$. Since both $0 \leq X_t \leq 1$ and $0 \leq \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z}) \leq 1$, we can bound $|Y_t|$ by 1:

$$|Y_t| = |X_t - \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z})| \leq \max(\mathcal{L}_{\mathrm{H}}(\boldsymbol{Z}), 1 - \mathcal{L}_{\mathrm{H}}(\boldsymbol{Z})) \leq 1.$$

Thus, we have established the uniform bound $|Y_t| \leq M$ with $M = 1$. Further, the variance of Y_t is identical to the variance of X_t, due to shift-invariance, meaning $\sigma_{Y_t}^2 \leq \frac{1}{4}$.

Finally, we apply Bernstein's inequality (Lemma A.4) to the sample mean with error threshold $\tau = \varepsilon$:

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{l=1}^{m} Y_{t_l}\right| \geq \varepsilon\right] \leq 2\exp\left(-\frac{m\varepsilon^2}{2\cdot\frac{1}{4} + \frac{2}{3}\cdot 1\cdot\varepsilon}\right) = 2\exp\left(-\frac{m\varepsilon^2}{\frac{1}{2} + \frac{2}{3}\varepsilon}\right).$$

Setting the right-hand side equal to the failure probability δ and solving for m yields the sample complexity bound stated in the theorem. □

A.4.3 Relation to the Wang–Wang and HypHC Costs

Our normalization uses the triplet base cost $\sum_t N_t^{\min}$ of Wang and Wang [2018]. Whereas their ratio cost divides by this base cost, we subtract it and divide by the triplet range. Using our rectified LCA depths in the continuous formulation of Chami et al. [2020] gives $C_{\mathrm{HypHC}}(\boldsymbol{Z}; w, \tau) = \sum_t N_t + 2\sum_{i<j} w_{ij}$. The following proposition shows that our normalization preserves its minimizers when the similarities are fixed.

Proposition 4.1 (Minimizer Equivalence under Fixed Similarities). Let the pairwise similarities $w = (w_{ij})$ be fixed, with $D(w) = \sum_t (N_t^{\max} - N_t^{\min}) > 0$, and let $\tau > 0$. Then $\mathcal{L}_{\mathrm{H}}(\,\cdot\,; w)$ is a strictly increasing affine transformation of $C_{\mathrm{HypHC}}(\,\cdot\,; w, \tau)$, and the two objectives have the same minimizers over any embedding domain.

Proof. By definition,

$$\mathcal{L}_{\mathrm{H}}(\boldsymbol{Z}; w) = \frac{C_{\mathrm{HypHC}}(\boldsymbol{Z}; w, \tau) - 2\sum_{i<j} w_{ij} - \sum_t N_t^{\min}}{D(w)}.$$

For fixed w, the subtracted terms and the positive denominator are constant in $\boldsymbol{Z}$. The transformation therefore preserves minimizers. □

Gradient through the LCA depths. With pairwise similarities held fixed, our normalization subtracts a constant from the continuous Dasgupta cost and divides by a positive constant. Its gradients with respect to the LCA depths are therefore scaled by the same positive factor, preserving their descent direction. Thus, the hierarchical loss retains the relaxation's preference for deeper LCAs between more similar pairs. During joint training, this signal acts alongside updates to the pairwise similarities themselves.

A.5 Datasets & Evaluation Metrics

A.5.1 Datasets

We evaluate on four image-classification benchmarks selected to cover a broad range of class counts and hierarchy depths.

CIFAR-100 [Krizhevsky et al., 2009]. 60,000 color images at 32×32 resolution across 100 fine-grained classes with 600 images each, split 50,000 / 10,000 for training and testing. We extend the dataset's canonical two-level (fine / coarse) labels into an 8-level hierarchy by mapping each fine class to its WordNet synset and traversing the hypernym chain to the WordNet root.

TinyImageNet [Le et al., 2015]. A subset of ImageNet-1k comprising 200 classes at 64×64 resolution, with 500 training and 50 validation images per class (100,000 / 10,000 total). Applying the same WordNet expansion as for CIFAR-100 yields a 7-level hierarchy.

ImageNet-1k [Deng et al., 2009]. 1,000 classes with 1,281,167 training and 50,000 validation images. The class hierarchy is the WordNet hypernym tree restricted to these 1,000 leaves, producing an 8-level structure. For tractability, we evaluate on a class-stratified 10,000-image subsample of the validation split.

iNaturalist-2018 [Van Horn et al., 2018]. 8,142 species with 437,513 training and 24,426 validation images. In contrast to the WordNet-based hierarchies above, iNaturalist-2018 supplies a biological taxonomy of depth 7 (kingdom → phylum → class → order → family → genus → species), which we adopt directly without WordNet supplementation. We evaluate on the full validation split. This dataset tests HyperTree on the largest number of leaf classes in our evaluation suite.

A.5.2 Evaluation Metrics

We report three families of metrics, derived from geodesic distances or rectified-LCA depths on each evaluation split, as specified below. For Euclidean baselines, geodesic distances reduce to Euclidean distances.

k-Nearest-Neighbor Accuracy. For each point $\boldsymbol{z}_i$ in the evaluation embeddings, we retrieve its k nearest neighbors (excluding itself) under the geodesic distance and take the most frequent fine-class label among them as the prediction. Accuracy at k is the fraction of evaluation points whose predicted label matches the ground-truth fine class. We report 1-NN accuracy in Appendix A.7.5.

Hierarchical Dendrogram Purity (hDP). Our headline metric hDP is the hierarchical generalization of dendrogram purity [Heller and Ghahramani, 2005]. It scores a binary dendrogram T over the evaluation points against the ground-truth hierarchy $\mathcal{H}$. For each pair of points (i, j) that share an ancestor at level ℓ of $\mathcal{H}$, we compute the purity of the subtree of T rooted at their lowest common ancestor, with respect to that ancestor's class set in $\mathcal{H}$. The final score is the pair-count-weighted average of these purities. hDP lies in $[0, 1]$, with 1 indicating perfect alignment between T and $\mathcal{H}$. We obtain T by agglomerative clustering using either pairwise geodesic distances or rectified-LCA depths, as specified below.

For comparisons across methods, we build T with `scipy.cluster.hierarchy.linkage` on the pairwise *geodesic*-distance matrix to retain the manifold structure. Since single, average, and complete linkage operate on pairwise dissimilarities they apply directly to geodesic distances. Ward's criterion, by contrast, is defined through within-cluster squared-Euclidean variance. Thus, its update rule over non-Euclidean geodesic distances does not retain this interpretation in general. For comparability, the results in Section 5 nonetheless fix Ward linkage as the common criterion and apply it identically to all methods and datasets. Section 5.4 reports hDP under all four criteria for the Poincaré Ball and Hyperboloid.

For HyperTree, we additionally evaluate the *LCA-induced tree*, defined by single-linkage agglomeration using the rectified-LCA depths as pairwise similarities, with larger depths indicating greater similarity. Since this decoder operates directly on the LCA depths that enter $\mathcal{L}_{\mathrm{H}}$, it provides the most direct readout of the learned hierarchy. Single linkage is fixed as part of the tree's definition rather than selected through tuning. We use this decoder to compare HyperTree across its own variants, while comparisons against external baselines (HIER, SeCu, SCAN) use Ward agglomeration over geodesic distances as described above.

Agglomerative Clustering NMI. In addition to 1-NN accuracy and hDP, we report Normalized Mutual Information (NMI) on a flat partition by cutting a Ward-linkage dendrogram into k clusters, with k set to the number of fine-grained classes in the evaluation set (100 for CIFAR-100, 200 for TinyImageNet, 1000 for ImageNet-1k, and 8142 for iNaturalist-2018). Similar to hDP, the dendrogram is built with `scipy.cluster.hierarchy.linkage` on the pairwise geodesic-distance matrix. The resulting partition is compared against the ground-truth fine labels. While 1-NN accuracy probes the immediate neighborhood of each point, NMI measures how well a flat partition of the whole embedding recovers the class structure.

A.6 Experiment Details

This section provides the full implementation details for the experiments reported in Section 5. The hyperparameter table at the end (Appendix A.6.6) summarizes the recipe numerically, while the surrounding subsections motivate each component.

A.6.1 Backbone

In our experiments, the choice of backbone and pretraining varies by dataset. For CIFAR-100 and TinyImageNet, we pretrain a ResNet-50 [He et al., 2016] backbone from scratch with SimCLR [Chen et al., 2020], using BYOL augmentations [Grill et al., 2020], InfoNCE temperature 0.5, learning rate 3×10^{-3}, and batch sizes of 1024 and 512, respectively, for 800 epochs. On ImageNet-1k and iNaturalist-2018, we instead build on public pretrained encoders that remain frozen throughout joint

training: a ResNet-50 trained with DINO [Caron et al., 2021] for ImageNet-1k, and the ViT-B/16 image encoder of BioCLIP [Stevens et al., 2024] for iNaturalist-2018. All inputs are processed at 224×224 resolution.

A.6.2 Projection Head and Manifold

The backbone features (2048-dimensional for ResNet-50; 512-dimensional for BioCLIP) pass through a Euclidean projection head that progressively reduces their dimensions (see *Projection Head, Euclidean dims* in Appendix A.6.6). Following feature scaling (Appendix A.2.5) the representations are then mapped into hyperbolic space via the exponential map at the origin, and a single hyperbolic linear layer (that depends on the hyperbolic manifold; see Appendix A.2.6) maps them into the final d-dimensional hyperbolic embedding space. Unless stated otherwise, all main-paper results use curvature magnitude $c = 1$. All hyperbolic operations are computed in float64 to mitigate the well-known numerical instabilities.

A.6.3 Pretraining

On CIFAR-100 and TinyImageNet, we pretrain the model with a purely Euclidean contrastive objective before joint training: $\mathcal{L}_{\mathrm{C}}$ with cosine similarity (Appendix A.3.1), applied to the outputs of the Euclidean projection head. This phase trains the backbone and the projection head from scratch. Since the loss is computed before the exponential map, no hyperbolic operation enters the objective, and the subsequent feature scaling and hyperbolic linear layer remain at their initialization. ImageNet-1k and iNaturalist-2018 skip this phase, since their encoders are already pretrained (DINO and BioCLIP, respectively). On all datasets, the model at the start of joint training serves as the epoch-0 anchor (Appendix A.6.5).

A.6.4 Joint Training Schedule

Starting from the pretrained model on CIFAR-100 and TinyImageNet (Appendix A.6.3), or from the frozen pretrained encoders on ImageNet-1k and iNaturalist-2018, we optimize $\mathcal{L}_{\text{total}} = \lambda_{\mathrm{C}}\,\mathcal{L}_{\mathrm{C}} + \lambda_{\mathrm{H}}\,\mathcal{L}_{\mathrm{H}}$ with Adam (learning rate 1×10^{-4}, weight decay 1×10^{-5}) for a dataset-dependent number of epochs (Appendix A.6.6). The projection head, feature scaling, and hyperbolic linear layer are trained on all datasets, whereas the backbone is trained only on CIFAR-100 and TinyImageNet and stays frozen on ImageNet-1k and iNaturalist-2018. The contrastive weight is held constant at $\lambda_{\mathrm{C}} = 0.1$ throughout, while the hierarchical loss weight λ_{H} is linearly increased from 1 to 5 over the first 250 epochs of joint training (or its full duration, if shorter). We keep the contrastive loss active to prevent the hierarchical loss from stalling training in boundary-saturated configurations. For the same reason, λ_{H} is not applied at full weight from the start, such that the contrastive term continues to shape the embedding as the hierarchical constraint tightens. Over the same interval, we linearly anneal the soft-LCA temperature of the hierarchical loss from $\tau = 0.2$ to 0.1. The temperature controls how sharply σ_τ selects among the three pairs of a triplet. At higher τ, the gradient of each triplet is spread over all three pairs rather than concentrated on the pair that currently has the deepest LCA, which makes the early training signal less sensitive to merge orders that are still changing. As $\tau \to 0$, N_t approaches the discrete per-triplet cost of Dasgupta [2016], Wang and Wang [2018].

A.6.5 Baselines

We compare against three baseline methods. Among Euclidean unsupervised methods, we use SeCu [Qian, 2023] and SCAN [Van Gansbeke et al., 2020], both strong contrastive deep-clustering pipelines. As a supervised hyperbolic point of reference, we include HIER [Kim et al., 2023], the architecturally closest method to ours but trained with full fine-class labels. Since none of these baselines report hierarchical dendrogram purity in their original papers, we extract embeddings from each (released checkpoints for SeCu and SCAN, a local rerun for HIER) and feed them through exactly the same evaluation pipeline as our model, ensuring all reported numbers are produced by identical code. Finally, we report our own model at epoch 0 of joint training: the same encoder, projection head, and hyperbolic linear layer, evaluated through the identical pipeline before any joint training has taken place. The Δ from epoch 0 to the final joint-training epoch isolates the contribution of training itself, controlling for both encoder quality and head architecture.

A.6.6 Hyperparameters

Parameter	Value
General Training	
Seeds	3 random seeds per configuration; 1 for compute-heavy ablations
Optimizer	Adam
Numerical precision	float64
Backbone with Pretraining (CIFAR-100, TinyImageNet)	
Architecture	ResNet-50
Pretraining objective	SimCLR (InfoNCE, $\tau_{\text{contr}} = 0.5$)
Augmentations	BYOL augmentations
Pretraining epochs	800
Pretraining LR	3×10^{-3}
Batch size	1024 (CIFAR-100); 512 (TinyImageNet)
Backbone (ImageNet-1k, iNaturalist-2018)	
Architecture	ResNet-50 (ImageNet-1k); ViT-B/16 (iNaturalist-2018)
Pretraining	DINO (ImageNet-1k); CLIP contrastive image-text objective (iNaturalist-2018)
Batch size	256 (ImageNet-1k); 256 (iNaturalist-2018)
Projection Head	
Euclidean dims	$[1024, 512, 256, 128]$ (CIFAR-100); $[1024, 512, 256]$ (TinyImageNet, ImageNet-1k); $[256]$ (iNaturalist-2018)
Manifold	Poincaré Ball (ablation default); Hyperboloid
Curvature magnitude (c)	1.0
Embedding dim (d)	16 (CIFAR-100, TinyImageNet); 32 (ImageNet-1k); 128 (iNaturalist-2018)
Hyperbolic linear	manifold-specific (Appendix A.2.6)
Joint Training Phase	
Epochs	300 (CIFAR-100, TinyImageNet); 50 (ImageNet-1k, iNaturalist-2018)
Learning rate	1×10^{-4}
Weight decay	1×10^{-5}
Backbone	trainable (CIFAR-100, TinyImageNet); frozen (ImageNet-1k, iNaturalist-2018)
Projection head	trainable
Contrastive weight λ_{C}	0.1 (held constant)
Hierarchical weight λ_{H}	Linear increase $1 \to 5$ over first 250 epochs (or full duration if shorter)
Soft-LCA temperature τ	Linear anneal $0.2 \to 0.1$ over first 250 epochs (or full duration if shorter)
Triplet Sampling (Theorem 4.5)	
Target deviation ε	0.01
Failure probability δ	0.01 (confidence $1 - \delta = 0.99$)
Triplets per step	$\sim$27,000

Table 10: Hyperparameters for all main-paper experiments.

A.7 Ablation Studies

This section collects ablations of the individual components of HyperTree. All HyperTree ablations in this section use the **Poincaré Ball**. The linkage and manifold ablations are presented in the main paper (Sections 5.3.3 and 5.4). We study, in turn, the training dynamics and loss-component decomposition (Appendices A.7.1 and A.7.2), latent dimensionality (Appendix A.7.3), depth decomposition (Appendix A.7.4), fine-grained retrieval and flat-clustering NMI (Appendix A.7.5), and iNaturalist curvature (Appendix A.7.6). Unless stated otherwise, ablations that compare HyperTree across its own variants report hDP on the LCA-induced tree, while comparisons against external baselines (HIER, SeCu, SCAN) report hDP under Ward linkage (Appendix A.5.2).

A.7.1 Loss Dynamics and Interplay

Figure 4 shows the training dynamics on all four datasets. For each dataset, we report the unweighted contrastive and hierarchical terms, the weighted total loss being optimized, and the hDP of the LCA-induced tree on the evaluation split. Over the course of joint training, the hierarchical loss decreases on every dataset, whereas the behavior of the contrastive loss depends on whether the backbone is updated. With a frozen backbone (ImageNet-1k, iNaturalist-2018), the contrastive loss decreases steadily. With a jointly trained backbone (CIFAR-100, TinyImageNet), it first decreases, then rises temporarily as λ_{H} increases, and ends training below its earlier minimum. This indicates that the two terms compete transiently while the backbone adapts to the hierarchical objective. On all datasets, most of the gain in hDP occurs early in joint training, after which it improves marginally or plateaus and stabilizes.

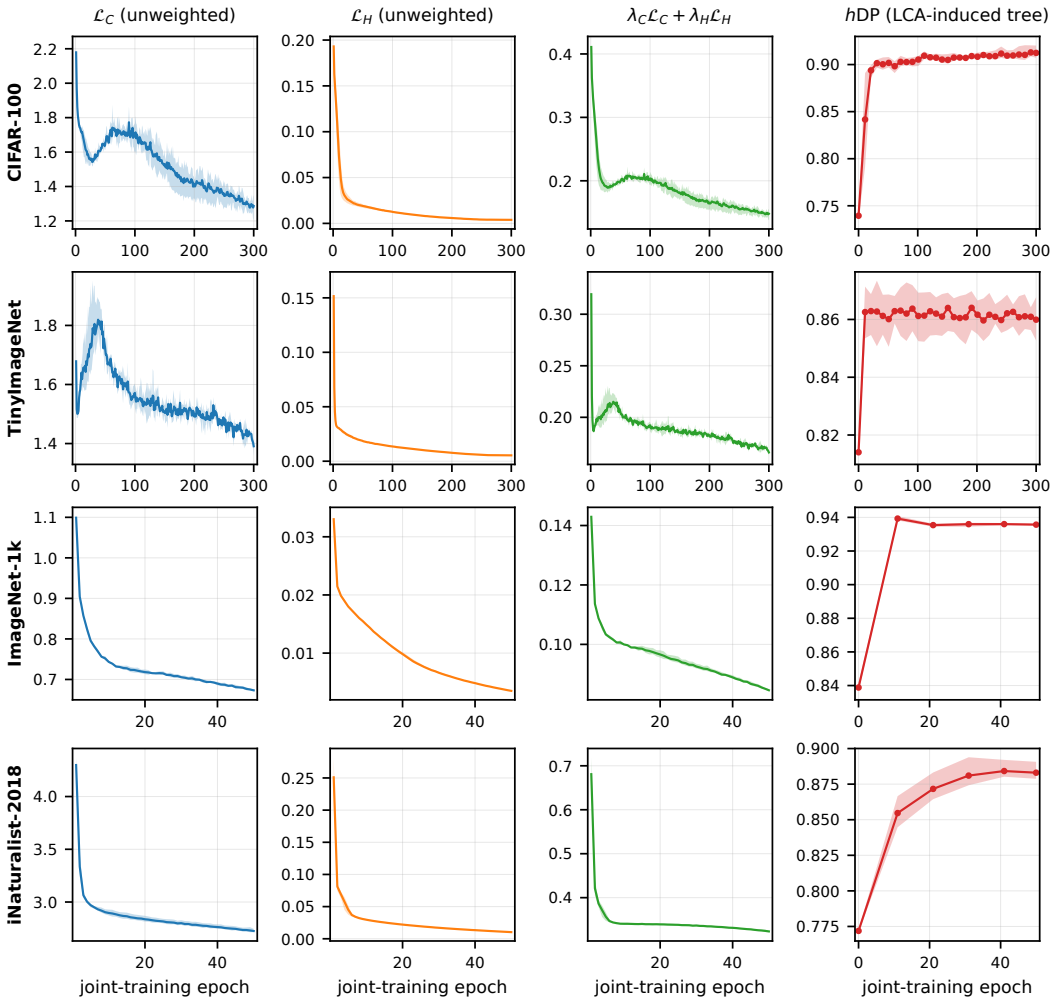

Figure 4: Loss dynamics during joint training. We show the unweighted contrastive loss $\mathcal{L}_{\mathrm{C}}$, the unweighted hierarchical loss $\mathcal{L}_{\mathrm{H}}$, and the optimized weighted total loss $\lambda_{\mathrm{C}}\mathcal{L}_{\mathrm{C}} + \lambda_{\mathrm{H}}\mathcal{L}_{\mathrm{H}}$. The hDP of the LCA-induced tree is evaluated on the evaluation split at the start of joint training and every ten epochs thereafter. Lines indicate the mean over three seeds, and shaded bands the min–max range. Where no band is clearly visible, the seed spread is narrower than the line width.

A.7.2 Loss-Component Decomposition

To separate the contributions of the contrastive and hierarchical loss terms, we compare contrastive-only ($\lambda_H = 0$), hierarchical-only ($\lambda_C = 0$), and joint training (Appendix A.6.4). The configurations share the pretrained backbone, head architecture, and training schedule of Appendix A.6.6. Each active loss retains its scheduling recipe: $\lambda_C = 0.1$, or the linear increase of λ_H from 1 to 5. Table 11 reports hDP on the LCA-induced tree across all variants, together with the epoch-0 anchor.

Configuration	CIFAR-100	TinyImageNet	ImageNet-1k
Epoch-0 anchor	0.735	0.813	0.837
Contrastive-only	0.846	0.834	0.871
Hierarchical-only	0.733	0.825	0.843
Joint training	$\mathbf{0.912 \pm 0.007}$	$\mathbf{0.860 \pm 0.008}$	$\mathbf{0.936 \pm 0.001}$

Table 11: Loss-component decomposition: hDP on the LCA-induced tree for contrastive-only, hierarchical-only, and joint training under otherwise identical recipes, compared against the epoch-0 anchor. Ablations and the anchor use seed 1; joint training reports mean $\pm$ standard deviation over three seeds, including seed 1, as in Table 2. **Bold** marks the best result per dataset.

A.7.3 Effect of Latent Dimensionality

Table 12 compares HyperTree with HIER and SeCu at latent dimensionalities $d \in \{16, 32, 128\}$. At its default dimensionality, HyperTree outperforms every baseline across all datasets where results are available. With a frozen backbone (ImageNet-1k, iNaturalist-2018), HyperTree's hDP under Ward linkage is largely insensitive to the choice of latent dimension d, while HIER depends strongly on it. With a jointly trained backbone (CIFAR-100, TinyImageNet), increasing d does not consistently benefit HyperTree under a training recipe tuned for low dimensionality. Overall, these results suggest that $d = 16$–32 suffices for strong hierarchical recovery, even for a taxonomy as large as that of iNaturalist-2018. Finally, since non-default HyperTree dimensionalities are evaluated with a single seed and without hyperparameter retuning, they do not rule out gains at higher dimensionality.

Dataset	Method	Unsup.	$d = 16$	$d = 32$	$d = 128$
CIFAR-100	SeCu	✓	0.416 ± 0.011	0.415 ± 0.012	0.380 ± 0.038
	HIER	✗	0.797 ± 0.025	$\mathbf{0.797 \pm 0.013}$	$\mathbf{0.769 \pm 0.012}$
	HyperTree	✓	$\mathbf{0.898 \pm 0.008}$	0.785	0.758
TinyImageNet	HIER	✗	0.447	0.425	$\mathbf{0.848 \pm 0.002}$
	HyperTree	✓	$\mathbf{0.862 \pm 0.006}$	**0.871**	0.831
ImageNet-1k	HIER	✗	0.544 ± 0.023	0.531 ± 0.004	0.897 ± 0.003
	HyperTree	✓	**0.936**	$\mathbf{0.934 \pm 0.000}$	**0.934**
iNaturalist-2018	HyperTree	✓	**0.928**	**0.925**	$\mathbf{0.921 \pm 0.009}$

Table 12: Effect of latent dimensionality on hDP under Ward linkage. We report mean $\pm$ standard deviation over three seeds where available and single-seeded results otherwise. **Bold** marks the best result per dataset and dimensionality.

A.7.4 Depth-Decomposed hDP on CIFAR-100

To examine how hierarchical recovery varies across different levels of the taxonomy, we evaluate hDP on CIFAR-100 separately at each depth of the WordNet hierarchy, excluding the root. Table 13 reports results under Ward linkage at $d = 16$, with margins computed against depth-specific random floors (random label permutations). HyperTree exceeds both the random floor and the epoch-0 anchor at every depth. The training gains are largest at intermediate depths and diminish towards the finest levels. Relative to HIER, HyperTree achieves larger floor-adjusted margins at depths 1–4, whereas HIER performs better at depths 5–7.

Depth	Size (count)	HyperTree hDP	Margin	Δ_{Ward}	HIER margin
1	25.0 (4)	0.853	+0.308	+0.227	+0.113
2	11.5 (8)	0.839	+0.384	+0.325	+0.175
3	8.6 (9)	0.867	+0.446	+0.391	+0.182
4	5.7 (11)	0.541	+0.389	+0.306	+0.222
5	5.1 (7)	0.308	+0.236	+0.148	+0.256
6	2.6 (8)	0.260	+0.225	+0.080	+0.424
7	2.0 (2)	0.285	+0.264	+0.045	+0.461

Table 13: Depth-wise hDP under Ward linkage on CIFAR-100. Depths range from coarsest (1) to finest (7), excluding the root. Size denotes the mean number of classes per cluster, with the number of clusters in parentheses. Margin denotes hDP gains against the depth-specific random floor (averaged over five label permutations), and Δ_{Ward} the gain over the epoch-0 anchor. All model entries use seed 1. The same depth-specific random ground-truth floor is used for all methods. HIER results are reported as a margin over the random floor.

A.7.5 Nearest-Neighbor and Flat-Clustering Results

To complement hierarchical dendrogram purity, we evaluate fine-class discriminability using 1-NN accuracy (Table 14) and flat-clustering quality using agglomerative-clustering NMI (Table 15). Among unsupervised methods, HyperTree achieves the highest scores on both metrics for CIFAR-100 and TinyImageNet, but trails SeCu on ImageNet-1k.

Methods are evaluated at their respective reported dimensionalities: $d = 16$ for HyperTree on CIFAR-100 and TinyImageNet, $d = 32$ for HyperTree on ImageNet-1k, and $d = 128$ on iNaturalist-2018. SCAN uses $d = 128$ on all three datasets. SeCu uses $d = 16$ on CIFAR-100 and $d = 128$ on TinyImageNet and ImageNet-1k. For iNaturalist-2018, the tables report three-seed results at the default curvature $c = 1.0$. Comparing seed 1 at both curvatures, reducing c from 1.0 to 0.3 increases 1-NN accuracy from 0.111 to 0.232 and geodesic-Ward NMI from 0.884 to 0.899.

Method	Unsup.	CIFAR-100	TinyImageNet	ImageNet-1k	iNaturalist-2018
SCAN	✓	0.412	0.037 ± 0.006	0.513	n/a
SeCu	✓	0.469	0.079 ± 0.039	$\mathbf{0.604}$	n/a
HyperTree	✓	$\mathbf{0.534 \pm 0.003}$	$\mathbf{0.237 \pm 0.004}$	0.235 ± 0.005	$\mathbf{0.113 \pm 0.004}$

Table 14: Fine-class 1-NN accuracy. We report mean $\pm$ standard deviation over three seeds where available and single-seeded results otherwise. **Bold** marks the best unsupervised result per dataset.

Method	Unsup.	CIFAR-100	TinyImageNet	ImageNet-1k	iNaturalist-2018
SCAN	✓	0.504	0.339 ± 0.008	0.722	n/a
SeCu	✓	0.584 ± 0.007	0.385 ± 0.064	$\mathbf{0.777}$	n/a
HyperTree	✓	$\mathbf{0.632 \pm 0.001}$	$\mathbf{0.488 \pm 0.001}$	0.736 ± 0.001	$\mathbf{0.884 \pm 0.001}$

Table 15: Fine-class clustering NMI under Ward linkage using geodesic distances. We report mean $\pm$ standard deviation over three seeds where available and single-seeded results otherwise. **Bold** marks the best unsupervised result per dataset.

A.7.6 Curvature Ablation on iNaturalist-2018

Table 16 compares HyperTree on iNaturalist-2018 across curvature magnitudes $c \in \{0.1, 0.3, 1.0\}$, while fixing all remaining settings. The optimal curvature depends on the decoder: c=1.0 yields the highest hDP on the LCA-induced tree, whereas c=0.3 performs best under Ward linkage. We use c=1.0 for all other iNaturalist-2018 experiments, since the LCA-induced tree is the target for comparing HyperTree's own variants.

	LCA-induced tree			**Ward linkage**		
Curvature (c)	**epoch 0**	**trained**	$\mathbf{\Delta_{LCA}}$	**epoch 0**	**trained**	$\mathbf{\Delta_{Ward}}$
0.1	0.772	0.814	+0.043	0.856	0.933	+0.077
0.3	0.772	0.835	+0.063	0.856	**0.938**	+0.081
1.0	0.772	**0.880**	+0.108	0.856	0.927	+0.071

Table 16: Effect of curvature on iNaturalist-2018 ($d = 128$). We report hDP for the LCA-induced tree and Ward linkage before (epoch 0) and after joint training, using seed 1 for every curvature. Deltas are computed from unrounded values before rounding. The three-seed default-curvature results are reported separately in Table 2. **Bold** marks the best result per tree decoder.

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: Section 4 presents the rectified LCA, hierarchical loss, and sampling bound. Table 2 reports results on four datasets, with external baselines on three.

   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: Section 6 discusses reliance on pretrained encoders. Appendices A.7.3, A.7.5 and A.7.6 examine weaker fine-grained recovery and sensitivity to dimension and curvature.

   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: [Yes]

Justification: Appendix A.4 collects the assumptions and proofs for the LCA projections, the loss-estimation bound, and the equivalence at fixed similarities.

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: Appendix A.6 collects all implementation details and hyperparameter settings.

   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: Our code is released at https://github.com/univie-dm/HyperTree. The hyperbolic-geometry primitives are provided by the publicly released `hyperbolic_math` library [Lang et al., 2026] (`https://doi.org/10.5281/zenodo.19236209`), used as an external dependency. Appendix A.5.1 lists the sources of all four public benchmarks.

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: Appendices A.5.1 and A.5.2 specify the evaluation splits, metrics, and decoders. Appendix A.6.4 and table 10 give the optimizer, schedules, batch sizes, and triplet budget.

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: Entries with $\pm$ give the mean and standard deviation over three seeds. Single-seed ablations use seed 1 (Tables 11 and 16). Shaded bands in Figure 4 span the minimum and maximum across seeds.

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: [No]

   Justification: Table 9 reports sampler time per step, and Table 10 gives batch sizes and epochs. A full per-configuration compute breakdown is unavailable in the paper.

   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: We reviewed the NeurIPS Code of Ethics. Our experiments use existing image benchmarks and involve no human participants (Appendix A.5.1).

   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: [N/A]

    Justification: This work is foundational research on unsupervised hierarchical representation learning. Positive societal impact: removing the dependence on fine-grained class labels reduces both the annotation burden and the embedded bias of curated label hierarchies,

and the explicit hierarchical objective produces interpretable multilevel structure that is auditable by domain experts (e.g. for taxonomic data such as iNaturalist-2018). Negative societal impact: as with any general-purpose representation-learning advance, we consider the contribution of this work to such risks to be small or negligible relative to existing image-representation methods.

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: We provide research code for learning image embeddings and use existing benchmark datasets.

    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: We credit and cite the datasets, encoders, and baselines in Appendices A.5.1, A.6.1 and A.6.5, and the hyperbolic geometry library separately [Lang et al., 2026].
    Datasets. CIFAR-100 [Krizhevsky et al., 2009] (no formal license is posted; the

Toronto release page only requests that the original tech report be cited, which we do); TinyImageNet [Le et al., 2015] (Stanford CS231n release; no explicit license is posted with the distribution); ImageNet-1k [Deng et al., 2009] (used under the ImageNet-1k Terms of Access, `https://image-net.org/accessagreement`, which restrict use to non-commercial research and educational purposes); iNaturalist-2018 [Van Horn et al., 2018] (per-image Creative Commons licenses—CC0, CC BY, CC BY-NC, CC BY-NC-SA, etc.—declared in the dataset metadata, used in accordance with the iNaturalist Terms of Service as required by `https://github.com/visipedia/inat_comp/tree/master/2018`).
Encoders and pretraining. ResNet-50 [He et al., 2016] implementation from `torchvision` (BSD-3-Clause, `https://github.com/pytorch/vision`); SimCLR [Chen et al., 2020] (Apache-2.0, `https://github.com/google-research/simclr`); DINO [Caron et al., 2021] (Apache-2.0, `https://github.com/facebookresearch/dino`); BioCLIP ViT-B/16 checkpoint [Stevens et al., 2024] (MIT, `https://huggingface.co/imageomics/bioclip`).
External library. The `hyperbolic_math` library [Lang et al., 2026] (MIT, Zenodo DOI 10.5281/zenodo.19236209).
Baseline reference implementations cited in Section 3 and Table 2: HIER [Kim et al., 2023] (MIT, `https://github.com/tjddus9597/HIER-CVPR23`); SCAN [Van Gansbeke et al., 2020] (Creative Commons Attribution-NonCommercial 4.0 International, `https://github.com/wvangansbeke/Unsupervised-Classification`, used here only for non-commercial research reproduction); SeCu [Qian, 2023] (Apache-2.0, `https://github.com/idstcv/SeCu`).

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 repository README documents installation, training configurations, and evaluation for the released code.

    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 experiments use existing datasets and involve neither crowdsourcing nor human participants.

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: This work does not involve human subjects, so no IRB review or equivalent ethics approval was required.

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: The method and evaluation use no large language model.

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.