WHEN LABELS HAVE STRUCTURE: IMPROVING IMAGE CLASSIFICATION WITH HIERARCHY-AWARE CROSS-ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard cross-entropy is the default classification loss across virtually all of machine learning, yet it treats all misclassifications equally, ignoring the semantic distances that a class hierarchy encodes. We propose Hierarchy-Aware Cross-Entropy (HACE), a drop-in replacement for standard cross-entropy that incorporates a known class hierarchy directly into the loss. HACE combines two components: prediction aggregation, which propagates the model's probability mass upward through the class hierarchy so that parent nodes accumulate the confidence of their children; and ancestral label smoothing, which distributes the ground-truth signal along the path from the true class to the root. We evaluate HACE on CIFAR-100, FGVC Aircraft, and NABirds in two regimes: end-to-end training across six architectures spanning convolutional and attention-based designs, and linear probing on frozen DINOv2-Large features. In end-to-end training, HACE improves top-1 accuracy over standard cross-entropy in 15 out of 18 architecture–dataset pairs, with a mean gain of 5.40 percentage points. In linear probing, HACE outperforms all competing methods on all three datasets, with a mean improvement of 2.18 points over the next best baseline. Beyond leaf-level accuracy, HACE also makes better mistakes: its predictions are more accurate at coarser levels of the taxonomy. Evaluated with hierarchical metrics on NABirds, it achieves the best operating curves and the smallest distance to the ground truth in the taxonomy in 5 of 6 architectures.

1 INTRODUCTION

Image classification has driven much of the progress of modern machine learning. Convolutional networks (LeCun et al., 1998; Krizhevsky et al., 2012), residual connections (He et al., 2016), normalization and regularization techniques (Ioffe & Szegedy, 2015; Srivastava et al., 2014), and large-scale data augmentation (Shorten & Khoshgoftaar, 2019) were all developed to improve classification accuracy, and attention-based architectures (Dosovitskiy et al., 2021; Liu et al., 2021) and modernized convolutional designs (Liu et al., 2022) now define the state of the art.

Despite these advances, progress has been driven almost exclusively by improvements in architectures, optimization, and data. The classification objective itself has received far less attention, and standard cross-entropy remains the default choice in virtually every image classification pipeline. It treats all classes as independent, assigning equal penalty to every incorrect prediction, regardless of how semantically distant it is from the correct class. Yet this is rarely how we think about errors. A model that confuses a husky with a beagle has made a smaller mistake than one that confuses it with a Persian cat, since the first two are both dogs while the third belongs to a different species. Confusing a husky with a motorcycle is a bigger mistake still. The hierarchy that makes these distinctions meaningful is often known in advance, encoded in dataset taxonomies or biological ontologies, yet standard cross-entropy discards it entirely.

In this work, we revisit the classification objective and incorporate hierarchy directly into the loss. We propose **Hierarchy-Aware Cross-Entropy (HACE)** as a drop-in replacement for standard cross-entropy whenever a hierarchy is available. HACE modifies both sides of the cross-entropy: *prediction aggregation* propagates model probability mass upward so that the score at each node

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

reflects confidence in the entire subtree below it; and *ancestral label smoothing* constructs a soft ground-truth distribution by placing decreasing probability mass at each ancestor of the true leaf class. We show in Section 3 that neither component yields a proper hierarchical loss on its own when labels are available only at the leaf level—the standard setting in image classification—and that using them together resolves this.

Aggregating leaf probabilities into ancestor probabilities is a standard device for hierarchical inference (Deng et al., 2012; Valmadre, 2022). Cultrera di Montesano et al. (2026) used it as a *training* objective, over a softmax that spans all nodes of a cell ontology, in a setting where training labels exist at multiple levels of the hierarchy. We adapt this mechanism to the leaf-only supervision regime and pair it with ancestral label smoothing to provide a training signal at every level of the hierarchy.

We conduct experiments on three datasets with hierarchical label structures—CIFAR-100 (Krizhevsky, 2009), FGVC Aircraft (Maji et al., 2013), and NABirds (Van Horn et al., 2015)—in two regimes: end-to-end training across six architectures spanning convolutional and attention-based designs, and linear probing on top of frozen DINOv2-Large (Oquab et al., 2024) features. In end-to-end training, HACE improves top-1 accuracy over standard cross-entropy in 15 out of 18 architecture–dataset pairs, with a mean gain of 5.40 percentage points (pp). In linear probing, HACE outperforms all competing methods on all three datasets, with a mean improvement of 2.18 pp over the next best baseline. Against hierarchy-aware baselines—HXE and structure-aware soft labels (Bertinetto et al., 2020) and the soft-max-margin loss of Valmadre (2022)—as well as standard label smoothing, a HACE variant is the most accurate method in 5 of 6 architectures. Unlike many hierarchy-aware losses, which reduce mistake severity at the cost of top-1 accuracy (Bertinetto et al., 2020; Karthik et al., 2021), HACE improves both: its errors land closer to the true class in the taxonomy and it attains the best hierarchical operating curves in most configurations.

Our main contributions are:

- **Method.** HACE, a loss that encodes the class hierarchy into both the predicted and the target distribution. We establish why the two components must be used together under leaf-only supervision.
- **Leaf-level evaluation.** A large-scale evaluation covering six architectures, three datasets, two training regimes, and five competing objectives, showing that HACE improves top-1 accuracy in the large majority of configurations, and that standard label smoothing applied before ancestral smoothing brings further gains in end-to-end training.
- **Hierarchical evaluation.** An analysis of coarse-level accuracy, mistake severity, and hierarchical operating curves (Valmadre, 2022), showing that the gains of HACE are not limited to the leaves.

2 Related Work

Hierarchical classification and mistake severity. Early work on hierarchical classification relied on tree-structured or per-level classifiers (Silla Jr & Freitas, 2011), cost-sensitive objectives that penalize predictions in proportion to their distance in the taxonomy (Deng et al., 2010), or structured output models over label relation graphs (Deng et al., 2014); YOLO9000 (Redmon & Farhadi, 2017) factorizes a softmax into conditional probabilities along a WordNet-derived tree. Deng et al. (2010) and Bertinetto et al. (2020) argued that top-1 accuracy should be complemented by the *severity* of mistakes, measured through the height of the lowest common ancestor (LCA) between prediction and ground truth, and several studies have since reported that reducing mistake severity with hierarchy-aware losses tends to cost top-1 accuracy (Bertinetto et al., 2020; Karthik et al., 2021). Valmadre (2022) further proposed to evaluate hierarchical classifiers across all operating points, from coarse and safe to fine and risky, using hierarchical precision and recall (Kosmopoulos et al., 2015). We adopt both the mistake-severity and the operating-curve protocols.

Hierarchy in the predicted distribution. A first family of methods injects the hierarchy into the model's output distribution while keeping the target a one-hot vector. Summing the probabilities of the leaves below a node gives the probability of that node, which Deng et al. (2012) use to trade specificity for accuracy at inference time. HXE (Bertinetto et al., 2020) factorizes the leaf softmax into conditional probabilities at each level of the tree and weights the corresponding log-losses by a

factor that decays exponentially toward the root. This enriches the loss with hierarchical structure, but does not modify the target distribution: the ground truth remains a one-hot vector at the leaf. HACE differs in that it modifies both sides of the cross-entropy, providing a supervision signal at the internal nodes of the tree, not just at the leaf.Valmadre (2022) proposes *soft-max-descendant*, which places a cross-entropy term at every node on the path from the root to the ground truth and normalizes each against the nodes that are neither its ancestors nor its descendants; *soft-max-margin*, which adds a margin to the logits of all non-ancestors of the ground truth before a standard softmax; and PS softmax, which parameterizes each leaf logit as the sum of scores along its root-to-leaf path. Of these, soft-max-margin performs best, and we include it as a baseline alongside HXE. Closest to our work, Cultrera di Montesano et al. (2026) introduced a hierarchical cross-entropy for cell-type annotation that applies a softmax over *all* nodes of the cell ontology and aggregates predicted probabilities upward, so that predicting a fine-grained subtype implies endorsing all its ancestors. Because the training cells are annotated at multiple levels of the ontology (e.g. *T cell* and *CD4$^+$ T cell*), the targets are well-defined without modification. In image classification labels usually exist only at the leaves, so aggregation alone produces outputs at internal nodes but no training signal for them (Section 3.1); HACE closes this gap by also modifying the target.

Hierarchy in the target distribution. Label smoothing (Szegedy et al., 2016) replaces the one-hot target with a mixture of the true class and a uniform distribution, improving calibration and reducing overconfidence; Müller et al. (2019) show that it acts as a regularizer but can erase class structure from the penultimate representation. Structured variants distribute the smoothing mass according to class similarity: Bertinetto et al. (2020) assign each class a mass that decays exponentially with the height of its LCA with the ground truth, SimLoss (Kobs et al., 2020) uses a pairwise similarity matrix, and Peterson et al. (2019) derive soft labels from human annotator disagreement. All of these redistribute mass *horizontally* across leaf classes. Our ancestral label smoothing is *vertical*: it places mass only on the ancestors of the true class, with a geometric decay in depth, and it is compatible with any of the horizontal schemes above, which can be applied before it (Section 3.2).

Hierarchy in representations and at inference. Another line of work encodes the hierarchy in the representation rather than in the loss over classes: Barz & Denzler (2019) regress class embeddings whose cosine similarities reflect the taxonomy, Khrulkov et al. (2020) embed images in hyperbolic space, Garg et al. (2022) and Chang et al. (2021) learn level-specific features, and Zhang et al. (2022) use the hierarchy to define contrastive positives. These approaches require a pre-computed class embedding, a change of output geometry, or additional heads, whereas HACE only changes the loss. Finally, post-hoc methods act on an already trained classifier: Karthik et al. (2021) re-rank predictions by expected hierarchical cost, and Goren et al. (2024) retreat from the most probable leaf to coarser ancestors until a calibrated confidence threshold is met. They require no retraining and are complementary to HACE.

3 METHODS

Both standard cross-entropy (SCE) and HACE minimize the cross-entropy between a ground-truth distribution p and a predicted distribution q:

$$\mathcal{L} = -\sum_i p(i) \log q(i). \tag{1}$$

They differ only in how p and q are constructed. SCE uses a one-hot p over n leaf classes and a flat softmax q over the same n classes. HACE expands both to cover all N nodes in the class hierarchy.

We require a class hierarchy organized as a tree[1], with n leaf nodes and N total nodes, which we extract from each dataset's taxonomy as a directed tree with edges pointing from parent to child. We update q through *prediction aggregation* (Section 3.1), which propagates probability mass *upward* through the tree, and p through *ancestral label smoothing* (Section 3.2), which redistributes the ground-truth signal *along the ancestral path* of the true class. Neither modification is sufficient on its own in the leaf-only supervision regime; we explain why at the end of each subsection.

[1]The method extends to directed acyclic graphs (DAGs), since both components only depend on ancestor–descendant relationships; see Appendix A.3. All hierarchies in this paper are trees.

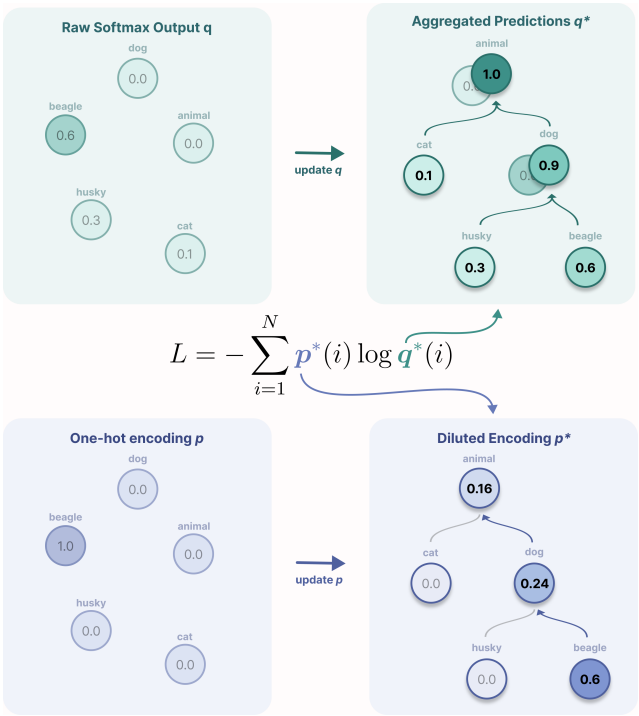

Figure 1: The two components of HACE on a toy animal hierarchy. **Prediction aggregation (updating q, top).** The raw softmax output q assigns probabilities to every node independently; aggregated predictions q^* are obtained by propagating mass upward through the tree (Eq. 2), so that each internal node accumulates the confidence of its entire subtree. **Ancestral label smoothing (updating p, bottom).** The one-hot target p places all mass on the ground-truth leaf; the ancestral (diluted) target p^* propagates mass upward along the path from the true class to the root (Eq. 4), here with $d = 0.6$ and $\varepsilon = 0$.

3.1 Prediction Aggregation

The model output layer is expanded to N units, one per node in the class hierarchy. Given raw logits $z \in \mathbb{R}^N$, we compute a softmax $\hat{q} = \mathrm{softmax}(z) \in \mathbb{R}^N$. Probability mass is then aggregated upward:

$$q^*(i) = \hat{q}(i) + \sum_{j \in \mathcal{D}(i)} \hat{q}(j), \tag{2}$$

where $\mathcal{D}(i)$ is the set of all descendants of node i. Thus $q^*(i)$ is the model's total probability that the true class is i or any of its subtypes, and $q^*(i) \geq q^*(j)$ for every descendant j of i. In practice $q^* = \mathcal{R}\hat{q}$ for a fixed binary reachability matrix $\mathcal{R}$ (Appendix A.1). If the taxonomy has a single root, $q^*(\text{root}) = 1$ by construction.

Why aggregation alone is not enough. With leaf-only supervision, the one-hot target p is zero at every internal node. The cross-entropy terms at those positions vanish regardless of q^*, so the aggregated outputs at internal nodes receive no gradient and the hierarchy is invisible to the loss.

3.2 Ancestral Label Smoothing

Existing soft-target methods modify p *horizontally*: they redistribute mass across leaf classes, either uniformly (Szegedy et al., 2016) or towards semantically close leaves (Bertinetto et al., 2020), so the target remains a distribution over leaves. Ancestral label smoothing instead propagates mass *vertically*, from the true leaf upward along its ancestral path, creating a supervision signal at every level of the hierarchy. It is this signal that the aggregated outputs of Section 3.1 are compared against: one component produces targets at internal nodes, the other produces model outputs at the same nodes.

Table 1: Summary of the three evaluation datasets. Hierarchical depth counts levels below the root, excluding the root itself.

Dataset	Train Set Size	Test Set Size	Leaf Classes	Total Classes	Hierarchical Depth	Cropped Image Size
CIFAR-100	50,000	10,000	100	120	2	32×32
FGVC	6,800	3,400	102	201	3	224×224
NABirds	23,929	24,633	555	1011	3 or 4	224×224

The two axes are complementary, and we combine them in two steps. Let a_0 be the ground-truth leaf class; we construct a soft target $p^* \in \mathbb{R}^N$ as follows.

Step 1: horizontal smoothing (optional). We first apply standard label smoothing to the one-hot target, with smoothing parameter $\varepsilon \in [0, 1)$:

$$\tilde{p}(i) = (1 - \varepsilon)\,\mathbf{1}[i = a_0] + \frac{\varepsilon}{n}, \quad i \in \{1, \ldots, n\}. \tag{3}$$

Setting $\varepsilon = 0$ recovers the one-hot target and restricts all smoothing to the vertical axis. The uniform distribution can be replaced by the LCA-based soft labels of Bertinetto et al. (2020).

Step 2: ancestral label smoothing (vertical). We introduce a dilution parameter $d \in (0, 1]$ and apply ancestral redistribution to the entire distribution $\tilde{p}$. For each leaf c with mass $\tilde{p}(c) > 0$, let $b_0, b_1, \ldots, b_k$ denote the path from c to the root, where $b_0 = c$. The contribution of leaf c to node b_j on its ancestral path is:

$$p^*(b_j) \mathrel{+}= d\,(1 - d)^j\,\tilde{p}(c) \quad \text{for } j < k, \qquad p^*(b_k) \mathrel{+}= \tilde{p}(c) - \sum_{j=0}^{k-1} d\,(1 - d)^j\,\tilde{p}(c), \tag{4}$$

where contributions from all leaves are summed at shared ancestors and all other nodes receive zero mass. Each node keeps a fraction d of the mass that reaches it and passes the rest to its parent, so most weight stays near the leaves; the last node on the path receives the remainder, and p^* sums to one. Note that a *larger* d means *less* dilution: when $d = 1$ no mass is propagated and p^* reduces to $\tilde{p}$. Figure 1 (bottom) illustrates p^* for $d = 0.6$ and $\varepsilon = 0$. In practice the targets are precomputed once per leaf class (Appendix A.2).

Why ancestral smoothing alone is not enough. Without prediction aggregation, a flat softmax over N nodes would treat a class and its own superset as competing alternatives: the target asks for mass on both *beagle* and *dog*, but increasing one decreases the other. This violates the basic constraint that a parent must be at least as probable as any of its children (Cultrera di Montesano et al., 2026); prediction aggregation enforces it by construction.

4 Experimental Setup

Datasets. We evaluate HACE on three datasets whose hierarchies differ in size, depth, and regularity (Table 1): CIFAR-100, a shallow and perfectly balanced two-level hierarchy; FGVC Aircraft, a moderately deep and unbalanced three-level hierarchy (manufacturer, family, variant); and NABirds, a biological taxonomy of variable depth. Details are given in Appendix B.1.

Models. We select six architectures spanning two design families: ResNet-18, ResNet-34, and ResNet-50 (He et al., 2016), to examine how model capacity interacts with the loss; ConvNeXt-T (Liu et al., 2022), a modernized convolutional design; and two attention-based models, ViT-B/16 (Dosovitskiy et al., 2021) and Swin-T (Liu et al., 2021). For linear probing, we use DINOv2-Large (Oquab et al., 2024), a ViT-L/14 trained with self-supervised objectives, as a frozen feature extractor. Details and hyperparameters are given in Appendix B.2.

Training procedure. In *end-to-end training*, all weights are optimized from random initialization on each dataset. In *linear probing*, each image is passed once through the frozen DINOv2-Large backbone, the unit-normalized classification token of the final layer is stored as a 1024-dimensional feature, and a single linear classifier is trained on top; since the classifier cannot learn new features, this setting isolates the effect of the loss from any interaction with feature learning. In both regimes we run every combination of architecture, dataset, and dilution value $d \in \{0.2, 0.5, 0.7\}$. Because ancestral label smoothing scales the leaf-level target, and therefore the leaf-level gradient, by d, we compare HACE and SCE under two learning-rate pairings that equalize the magnitude of leaf-level updates: one anchored at the HACE learning rate and one anchored at the SCE learning rate (Appendix B.3). This yields six HACE models and four SCE models per architecture–dataset pair. HACE and SCE follow identical forward passes, except that the HACE output layer produces N logits over all hierarchy nodes rather than n leaf logits. All other hyperparameters are set to the values recommended by the original authors and held fixed throughout, so that differences in performance can be attributed to the loss alone.

Leaf-level metrics. We report **top-1 accuracy**, the fraction of test samples whose highest-scoring leaf is the ground truth, in the main text, and top-5 accuracy in Appendix C.2. Both are computed over leaf predictions only, which makes HACE—whose output layer spans all N nodes—directly comparable with SCE.

Hierarchical metrics. Top-1 accuracy treats every mistake as equally wrong. To capture how far mistakes land from the truth, we report three additional evaluations. For all of them, internal-node probabilities are obtained by aggregating the model's probabilities upward (Eq. 2); HACE produces these natively, while for SCE and the baselines the aggregation of leaf probabilities is applied post-hoc at evaluation time. (i) **Coarse-level accuracy**: at a given internal level (e.g. FGVC families), the prediction is the node of that level with the highest aggregated probability, and it is correct if it is the ancestor of the true leaf. (ii) **Hierarchical distance**: the number of edges on the path between the predicted leaf and the true leaf in the taxonomy, averaged over all test samples (zero for correct predictions). (iii) **Operating curves** (Valmadre, 2022): sweeping a threshold on the aggregated probability, the model predicts the most specific node whose probability exceeds it, and each prediction is scored with information-based hierarchical precision and recall. We summarize the curves by their area (AP), by the area under the correct-vs-exact curve (AC), and by the exact recall at 90% and 95% correct (R@90C, R@95C), and report the information-based F1 both of the top-1 leaf prediction (F1 (Leaf)) and of the prediction at threshold 0.5, i.e. majority inference (F1 (Majority)).

5 Results

5.1 End-to-end training

Table 2 reports, for each architecture–dataset pair, the best top-1 accuracy achieved by HACE and SCE across dilution values and learning-rate pairings. HACE outperforms SCE in 15 out of 18 pairs, with a mean gain of 5.40 pp (median 2.61 pp) and a maximum decrease of 1.57 pp. The gains hold for both convolutional and attention-based designs. They are smallest on CIFAR-100 (mean 1.10 pp), whose hierarchy is shallow and balanced, and larger on FGVC (8.97 pp) and NABirds (6.14 pp), whose taxonomies are deeper and irregular.

Appendix Table 8 breaks the FGVC results down by dilution value and learning-rate pairing, comparing each HACE model with its learning-rate-matched SCE counterpart. HACE outperforms SCE in 32 of 36 configurations. Three SCE runs (ResNet-18 at $d = 0.7$ and ResNet-34 at $d \in \{0.5, 0.7\}$, HACE-anchored pairing) and one HACE run (Swin-T at $d = 0.2$, SCE-anchored pairing) diverged to near-chance accuracy. Excluding the three diverged SCE runs, HACE outperforms SCE by a mean of 4.20 pp, with a maximum gain of 21.79 pp.

5.2 Comparison with hierarchy-aware and smoothing baselines

To situate HACE among hierarchy-aware and smoothing-based objectives, we compare six methods in ten configurations on NABirds under identical training conditions (Table 3): HACE without and with standard label smoothing before the ancestral redistribution ($\varepsilon = 0.1$); SCE without and with

Table 2: Top-1 accuracy (%) of HACE and SCE in end-to-end training. Each cell reports the best result across dilution values $d \in \{0.2, 0.5, 0.7\}$ and both learning-rate pairings. The better result per pair is **bolded**.

	CIFAR-100		**FGVC**		**NABirds**	
	HACE	SCE	HACE	SCE	HACE	SCE
ResNet-18	**77.93**	76.52	**78.88**	68.29	**67.93**	62.94
ResNet-34	78.01	**78.30**	77.32	**77.74**	69.44	**71.01**
ResNet-50	**80.53**	79.86	**80.62**	69.88	**67.90**	59.07
ConvNeXt-T	**44.48**	43.30	**52.24**	30.45	**50.74**	30.35
ViT-B/16	**64.45**	63.56	**20.67**	15.72	**29.16**	27.44
Swin-T	**59.44**	56.72	**44.34**	38.16	**44.01**	41.51

Table 3: Top-1 accuracy (%) on NABirds under identical training conditions. HACE: prediction aggregation and ancestral label smoothing, without ($\varepsilon = 0$) and with ($\varepsilon = 0.1$) standard label smoothing before the ancestral redistribution. SCE: standard cross-entropy with one-hot targets, uniform label smoothing, or the LCA-based soft labels of Bertinetto et al. (2020). HXE (Bertinetto et al., 2020): hierarchically weighted cross-entropy. Margin: soft-max-margin (Valmadre, 2022) with margin m. The best result per architecture is **bolded** and the second best underlined.

	HACE		**SCE**				**HXE**		**Margin**	
	No smooth.	**Label smooth.**	**No smooth.**	**Label smooth.**	**Soft label $\beta = 10$**	**Soft label $\beta = 30$**	**$\alpha = 0.2$**	**$\alpha = 0.5$**	**$m = 2$**	**$m = 5$**
ResNet-18	67.93	**70.81**	62.94	66.57	67.04	59.21	61.73	65.98	57.06	56.2
ResNet-34	69.44	**73.02**	71.01	72.15	71.01	62.36	66.07	70.62	59.95	60.4
ResNet-50	67.90	**73.15**	59.07	71.47	70.41	59.98	63.06	69.57	57.33	56.3
ConvNeXt-T	50.74	**59.49**	30.35	40.33	0.94	1.00	0.59	0.47	32.31	34.1
ViT-B/16	**29.16**	17.53	27.44	23.14	15.64	17.50	18.78	17.72	18.01	19.0
Swin-T	44.01	47.10	41.51	47.27	**48.00**	42.00	43.06	44.29	43.02	44.9

uniform label smoothing (Szegedy et al., 2016); SCE with the LCA-based soft labels of Bertinetto et al. (2020); HXE (Bertinetto et al., 2020); and soft-max-margin (Valmadre, 2022).

A HACE variant is the most accurate method in 5 of 6 architectures, HACE with label smoothing in 4 of 6, and the best HACE variant exceeds the best competing method by 4.38 pp on average (median 1.70 pp; the mean is driven by ConvNeXt-T). The gap between HACE and SCE with label smoothing shows that the gains are not explained by smoothing alone. Adding standard label smoothing to HACE improves accuracy in 5 of 6 architectures, confirming that horizontal and vertical smoothing are complementary in end-to-end training; the exception is ViT-B/16. HXE and the soft labels fail to train on ConvNeXt-T under the author-recommended hyperparameters.

5.3 Hierarchical evaluation

Coarse-level accuracy. Using the ResNet-50 models of Table 2, HACE matches or outperforms SCE in per-class accuracy in 59 of 70 FGVC families and 28 of 30 manufacturers (mean gains of 7.21 and 4.21 pp), in 48 of 50 NABirds family/order groups, and in 16 of 20 CIFAR-100 superclasses. Figure 2 shows the per-class breakdown at the family level of FGVC: the largest gains are concentrated on the families where SCE is weakest. Per-class plots for all other levels are given in Appendix D.

Mistake severity and operating curves. Table 4 summarizes the hierarchical metrics of all methods on NABirds; the per-architecture results are in Appendix Table 13. HACE without label smoothing has the best operating curves: the highest mean AP, AC, R@90C, R@95C and F1 (Majority), and the best average rank on all of them except AC, where HACE with label smoothing ranks first. HACE with label smoothing has the highest F1 (Leaf) and the smallest hierarchical distance. Per architecture, a HACE variant is best on AP, AC, F1 (Leaf) and hierarchical distance in 5 of 6 architectures, on R@90C in 4 of 6, and on R@95C and F1 (Majority) in 3 of 6, losing mostly to HXE. Label smoothing trades the high-precision end of the operating curve (R@90C, R@95C, F1 (Majority)) for leaf accuracy: by lowering the confidence of the model it makes threshold-based predictions retreat to coarser nodes more often, which is the expected effect of horizontal smoothing on calibration (Müller et al., 2019).

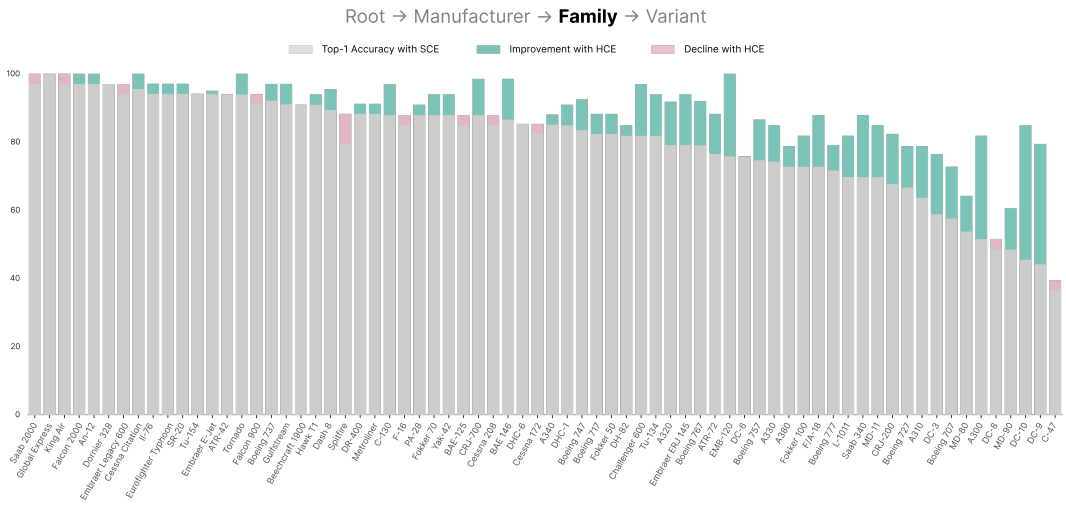

Figure 2: Per-class accuracy at the family level of FGVC Aircraft (ResNet-50). Grey bars show SCE accuracy; the colored segment shows the gain (teal) or decline (pink) with HACE. Classes are sorted by SCE accuracy. HACE matches or outperforms SCE in 59 of 70 families (mean gain 7.21 pp).

Table 4: Hierarchical metrics on NABirds (end-to-end), averaged over architectures; average rank across architectures in parentheses. AP: area under the information-precision vs. recall curve; AC: area under the correct vs. exact curve; R@90C/R@95C: exact recall at 90%/95% correct; F1 (Maj.): information-based F1 at threshold-0.5 (majority) inference; F1 (Leaf): information-based F1 of the top-1 leaf prediction; Dist.: mean hierarchical distance between the top-1 leaf and the true leaf (lower is better). For each method, the variant (dilution, learning-rate pairing, or hyperparameter) with the best AP is reported per architecture. The best mean per column is **bolded** and the second best underlined.

Method	AP ↑	AC ↑	R@90C ↑	R@95C ↑	F1 (Maj.) ↑	F1 (Leaf) ↑	Dist. ↓
HACE	**63.75** (2.5)	**50.06** (2.8)	**52.05** (1.3)	**45.16** (1.5)	**62.50** (1.7)	66.93 (3.3)	2.32 (3.3)
HACE + LS	62.03 (2.7)	49.95 (2.5)	42.25 (3.7)	38.04 (3.7)	43.56 (4.3)	**67.83** (1.8)	**2.25** (1.8)
SCE	56.26 (4.8)	43.92 (4.5)	36.43 (5.0)	32.51 (5.0)	35.61 (5.3)	62.83 (4.0)	2.58 (3.8)
SCE + LS	58.52 (3.3)	45.82 (3.0)	38.36 (4.2)	34.06 (4.2)	37.86 (4.8)	64.93 (2.5)	2.44 (2.5)
HXE	53.31 (2.8)	40.85 (3.8)	43.20 (2.5)	37.60 (2.3)	52.46 (2.5)	55.99 (3.8)	3.02 (4.0)
Margin	54.22 (4.8)	40.83 (4.3)	41.60 (4.3)	34.76 (4.3)	56.38 (2.3)	57.99 (5.5)	2.91 (5.5)

5.4 Linear probing

Table 5 reports the best top-1 accuracy of every method under linear probing on frozen DINOv2-Large features. HACE outperforms all other methods on all three datasets, with an average improvement of 2.18 pp over the next best (non-HACE) method. Unlike in end-to-end training, adding standard label smoothing does not help HACE in this regime. The per-dilution comparison with learning-rate-matched SCE (Appendix Table 9) is equally consistent: HACE outperforms SCE in 17 of 18 dataset–dilution configurations, with a mean gain of 1.59 pp (maximum 5.06 pp, maximum decrease 0.99 pp). The margins are smaller than in end-to-end training, as expected: all methods exceed 92% on CIFAR-100, leaving little room for improvement, while on the more challenging FGVC and NABirds the advantage of HACE is 4.87 pp and 1.46 pp.

6 Discussion and Conclusion

HACE consistently improves over standard cross-entropy across a diverse set of architectures, datasets, and training regimes, and, unlike many hierarchy-aware objectives, it does so on leaf accuracy and on hierarchical metrics at the same time. This supports our central claim: when a known hierarchy is

Table 5: Top-1 accuracy (%) under linear probing on frozen DINOv2-Large features. Each cell reports the best result across dilution values $d \in \{0.2, 0.5, 0.7\}$ and both learning-rate pairings. The best result per dataset is **bolded** and the second best underlined.

	HACE		SCE		HXE		Soft Label		Margin	
	No smooth.	Label smooth.	No smooth.	Label smooth.	$\alpha = 0.2$	$\alpha = 0.5$	$\beta = 10$	$\beta = 30$	$m = 2$	$m = 5$
CIFAR-100	**96.90**	96.81	96.62	96.68	92.72	92.52	92.90	92.95	92.93	92.98
FGVC	**58.97**	58.70	53.91	53.40	49.81	42.90	53.83	54.10	51.61	53.05
NABirds	**84.96**	84.78	83.50	83.02	79.09	64.86	83.00	83.50	78.80	81.92

available, incorporating it directly into the loss is a simple and effective way to improve classification, with no change to the architecture or the training pipeline and a negligible cost (a fixed $N \times N$ matrix product per batch).

Limitations. HACE requires a known class hierarchy, which may not always be available. The dilution d is an additional hyperparameter; while we show consistent gains across a fixed set of values, dedicated tuning could yield further improvements, and since d rescales the leaf-level gradient (Appendix B.3), it should ideally be tuned jointly with the learning rate. Our end-to-end experiments train all architectures from scratch with the authors' recommended hyperparameters, which leaves transformers in a data-starved regime on FGVC and NABirds; evaluating HACE when fine-tuning pretrained models, on larger taxonomies such as iNaturalist or ImageNet, and on ontologies that are DAGs rather than trees is left to future work. A theoretical characterization of optimal ancestral smoothing functions, and of the conditions under which they improve generalization, remains an open question.

AI USE STATEMENT

Large language model assistants were used to edit and polish the text, to check the consistency of reported numbers, and to help with LaTeX formatting. All research ideas, experiments, analyses, and conclusions are the authors' own, and the authors verified all content of the paper.

ETHICS STATEMENT

This work uses publicly available image classification benchmarks (CIFAR-100, FGVC Aircraft, NABirds) under their respective licenses and involves no human subjects or personal data. We do not foresee specific negative societal impacts beyond those common to image classification.

REPRODUCIBILITY STATEMENT

The loss is fully specified in Section 3, and its implementation through the reachability matrix and the precomputed target matrix is described in Appendices A.1 and A.2. Datasets, hierarchies, and preprocessing are described in Appendix B.1; architectures, hyperparameters, and epochs in Appendix B.2 and Table 6; and the learning-rate pairings in Appendix B.3.

REFERENCES

Björn Barz and Joachim Denzler. Hierarchy-based image embeddings for semantic image retrieval. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pp. 638–647. IEEE, 2019.

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12506–12515, 2020.

Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your "flamingo" is my "bird": Fine-grained, or not. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

S. Cultrera di Montesano, D. D'Ascenzo, S. Raghavan, A.P. Amini, P.S. Winter, and L. Crawford. Improving atlas-scale single-cell annotation models with hierarchical cross-entropy loss. *Nature Computational Science*, 6:243–249, 2026.

Jia Deng, Alexander C. Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In Kostas Daniilidis, Petros Maragos, and Nikos Paragios (eds.), *Computer Vision – ECCV 2010*, pp. 71–84. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15555-0.

Jia Deng, Jonathan Krause, Alexander C. Berg, and Li Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Computer Vision – ECCV 2014*, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Ashima Garg, Depanshu Sani, and Saket Anand. Learning hierarchy aware features for reducing mistake severity. In *Computer Vision – ECCV 2022*, 2022.

Shani Goren, Ido Galil, and Ran El-Yaniv. Hierarchical selective classification. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR.

Shyamgopal Karthik, Ameya Prabhu, Puneet K. Dokania, and Vineet Gandhi. No cost likelihood manipulation at test time for making better mistakes in deep networks. In *International Conference on Learning Representations*, 2021.

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*, pp. 6418–6428, 2020.

Konstantin Kobs, Michael Steininger, Albin Zehe, Florian Lautenschlager, and Andreas Hotho. Simloss: Class similarities in cross entropy. In *Foundations of Intelligent Systems: 25th International Symposium, ISMIS 2020, Graz, Austria, September 23–25, 2020, Proceedings*, pp. 431–439. Springer-Verlag, 2020.

Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3), 2015.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

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

Subhransu Maji, Juho Kannala, Esa Rahtu, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9617–9626, 2019.

Joseph Redmon and Ali Farhadi. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Carlos N Silla Jr and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data mining and knowledge discovery*, 22(1):31–72, 2011.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Jack Valmadre. Hierarchical classification at multiple operating points. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 18034–18045. Curran Associates, Inc., 2022. doi: 10.52202/068431-1311. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/727855c31df8821fd18d41c23daebf10-Paper-Conference.pdf.

Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 595–604, 2015.

Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. Use all the labels: A hierarchical multi-label contrastive learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

A METHOD DETAILS

A.1 REACHABILITY MATRIX

Both prediction aggregation and loss computation rely on the *reachability matrix* $\mathcal{R} \in \{0,1\}^{N \times N}$, where N is the total number of nodes in the class hierarchy (leaves and internal nodes combined):

$$\mathcal{R}_{ij} = \begin{cases} 1 & \text{if node } j \text{ is a descendant of node } i, \text{ or } j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Equivalently, $\mathcal{R}$ is the reflexive transitive closure of the parent-to-child adjacency matrix of the hierarchy: it encodes, for each node i, the set of nodes whose raw softmax probability is accumulated into $q^*(i)$. The aggregated prediction of Section 3.1 is then a single matrix–vector product, $q^* = \mathcal{R}\,\hat{q}$, which requires no explicit tree traversal and through which gradients flow back to the raw logits.

Properties. The reachability relation encoded in $\mathcal{R}$ is a partial order: it is reflexive, antisymmetric (the hierarchy is acyclic), and transitive. A direct consequence is that $q^*(i) \geq q^*(j)$ whenever j is a descendant of i, which is exactly the hierarchical consistency constraint.

Construction. For each dataset, $\mathcal{R}$ is built once before training by computing the transitive closure of the parent-to-child adjacency matrix and setting the diagonal to 1. At the scales considered here, the matrix fits comfortably in memory as a dense tensor and can be moved to GPU for batched products.

A.2 ANCESTRAL LABEL SMOOTHING MATRIX

To avoid recomputing the soft target p^* at every training step, we precompute it for every leaf class and store the result as the *ancestral label smoothing matrix* $\mathcal{T} \in \mathbb{R}^{n \times N}$, whose row i is the soft target p^* for leaf class i, constructed following Section 3.2 with the chosen ε and d. During training, the targets of a batch are retrieved by indexing, $P^*_{\text{batch}} = \mathcal{T}[\mathbf{y}] \in \mathbb{R}^{B \times N}$, where B is the batch size and $\mathbf{y} \in \{1, \ldots, n\}^B$ the vector of ground-truth leaf indices, a single gather operation per batch.

A.3 EXTENSION TO DIRECTED ACYCLIC GRAPHS

Section 3 assumes a tree, so that every node has a unique parent and the ancestral path from any leaf to the root is unique. Both components of HACE extend to directed acyclic graphs (DAGs) with a single root, where a node may have multiple parents.

Prediction aggregation requires no modification: $\mathcal{R}_{ij} = 1$ whenever j is reachable from i by following directed edges, $q^* = \mathcal{R}\,\hat{q}$ still accumulates the probability of all descendants, and $q^*(i) \geq q^*(j)$ for any descendant j of i. Note, however, that q^* is no longer a distribution over each level, since a node with several parents contributes to each of them.

Ancestral label smoothing requires splitting mass among parents. Each node keeps a fraction d of the mass that reaches it and passes the remaining fraction $(1-d)$ upward, split equally among its $|\mathcal{P}(v)|$ parents, where $\mathcal{P}(v)$ denotes the set of parents of v; the root keeps all the mass that reaches it. For a leaf c and a path $b_0 = c, b_1, \ldots, b_m = v$ from c to a non-root node v, the contribution along that path is

$$p^*(v) \mathrel{+}= d\,(1-d)^m \prod_{j=0}^{m-1} \frac{1}{|\mathcal{P}(b_j)|}\, \tilde{p}(c), \tag{6}$$

where the product accounts for the splitting at each node that passes mass upward. Summing over all paths from c to v and all leaves c with nonzero $\tilde{p}(c)$ yields the full soft target $p^* \in \mathbb{R}^N$, which remains a valid probability distribution: at each node the outgoing mass is partitioned, not duplicated, across parents. On a tree, $|\mathcal{P}(b_j)| = 1$ and Equation (6) reduces to Equation (4).

B EXPERIMENTAL DETAILS

B.1 DATASETS

CIFAR-100 (Krizhevsky, 2009). CIFAR-100 contains 60,000 colour images at 32×32 pixels, split into 50,000 training and 10,000 test samples. The 100 leaf classes are grouped into 20 superclasses of exactly 5 leaves each, giving a perfectly balanced two-level hierarchy (superclass $\rightarrow$ class): every leaf is at the same depth and every internal node has the same branching factor. This makes CIFAR-100 a clean testbed for measuring whether HACE helps when the hierarchy is simple and the images are low-resolution.

FGVC Aircraft (Maji et al., 2013). FGVC Aircraft contains 10,200 images of aircraft at 224×224 pixels. The label hierarchy has three levels below the root: *manufacturer* (e.g. Boeing), *family* (e.g. Boeing 737), and *variant* (e.g. Boeing 737-300). The original taxonomy defines 214 nodes; we exclude 13 internal nodes that have no associated images, yielding 201 nodes in total. The hierarchy is moderately deep and unbalanced: manufacturers have different numbers of families, and families differ in the number of variants they contain.

NABirds (Van Horn et al., 2015). NABirds contains approximately 48,000 images of North American birds at 224×224 pixels, split into 23,929 training and 24,633 test images. The 555 leaf classes are organized into a biological taxonomy. Most leaves follow a three-level path below the root, *order* $\rightarrow$ *species* $\rightarrow$ *variant*; the order *Passeriformes* (perching birds), which accounts for over 60% of all species, is further subdivided by *family*, resulting in a four-level path (order $\rightarrow$ family $\rightarrow$ species $\rightarrow$ variant). This mixture of depths makes NABirds the most irregular hierarchy in our evaluation: branching factors vary widely, leaf depth is not uniform, and the taxonomy mirrors a real biological ontology rather than a purpose-built classification scheme.

B.2 MODELS

Table 6 reports the hyperparameters of each architecture and the number of training epochs per dataset. All hyperparameters are set to the values recommended by the original authors and held fixed across loss functions and datasets; the only exception is the learning rate, which is adjusted as described in Appendix B.3.

Convolutional architectures. **ResNet-18, ResNet-34, ResNet-50** (He et al., 2016) are residual networks trained with SGD and cosine annealing; the three depths allow us to examine whether the benefit of HACE interacts with model capacity within one family. **ConvNeXt-T** (Liu et al., 2022) is a modernized convolutional design that borrows design choices from transformers (depthwise convolutions, layer normalization, inverted bottlenecks) while remaining fully convolutional; it is trained with AdamW.

Attention-based architectures. **ViT-B/16** (Dosovitskiy et al., 2021) splits each image into 16×16 patches and processes them with global self-attention; it uses dropout ($p = 0.1$) and is trained with AdamW. **Swin-T** (Liu et al., 2021) restricts self-attention to shifted local windows and progressively merges patches, giving a hierarchical feature map similar to a convolutional backbone; it is trained with AdamW.

Linear probing backbone. **DINOv2-Large** (Oquab et al., 2024) is a ViT-L/14 ($\sim$300M parameters) trained with self-supervised objectives on a large curated dataset. For each image, we extract the unit-normalized classification token from the final hidden state, producing a fixed 1024-dimensional representation, computed once per dataset before any training.

B.3 LEARNING RATES

Ancestral label smoothing sets the target mass of the true leaf to d (when $\varepsilon = 0$) rather than 1. Because the cross-entropy gradient with respect to the logit of the true class scales with the target value at that class, the leaf-level gradient under HACE is d times that under SCE for the same model output. If both losses were trained at the same learning rate ℓ, the leaf-level updates would differ by

Table 6: Hyperparameters for the six end-to-end architectures. Epochs are reported as CIFAR-100 (C) / FGVC (F) / NABirds (N).

Model	Type	Params	Batch	LR ℓ	Weight Decay	Momentum	Dropout	Epochs (C / F / N)	LR Scheduler	Optimizer
ResNet-18	Conv	∼11.7M	128	0.1	10^{-4}	0.9	—	450 / 750 / 750	CosineAnnealing	SGD
ResNet-34	Conv	∼21.8M	128	0.1	10^{-4}	0.9	—	450 / 750 / 750	CosineAnnealing	SGD
ResNet-50	Conv	∼25.6M	128	0.1	10^{-4}	0.9	—	450 / 750 / 750	CosineAnnealing	SGD
ConvNeXt-T	Conv	∼28.6M	128	1.25×10^{-4}	0.05	0.9	—	400 / 400 / 500	CosineAnnealing	AdamW
ViT-B/16	Attn	∼86M	64	3×10^{-3}	0.3	0.9	0.1	300 / 300 / 300	Lambda	AdamW
Swin-T	Attn	∼28.3M	128	1.25×10^{-4}	0.05	0.9	—	500 / 500 / 500	Lambda	AdamW

a factor of d, a confound we control for[2]. We therefore consider two symmetric pairings (Table 7), both of which train HACE at a learning rate $1/d$ times that of SCE:

- **HACE-anchored pairing (standard).** HACE is trained at the base rate ℓ recommended by the original authors, and SCE at $\ell \cdot d$, scaling its learning rate *down* so that its leaf-level updates match those of HACE.
- **SCE-anchored pairing (adj).** SCE is trained at the base rate ℓ, and HACE at ℓ/d, scaling its learning rate *up* to match the larger leaf-level gradient of SCE.

Table 7: Learning-rate pairings. The base learning rate ℓ is the value recommended by the original authors for each architecture (Table 6).

Pairing	Loss	Learning Rate
HACE-anchored (standard)	HACE	ℓ
HACE-anchored (standard)	SCE	$\ell \cdot d$
SCE-anchored (adj)	HACE	ℓ/d
SCE-anchored (adj)	SCE	ℓ

Each experiment is one triple (architecture, dataset, dilution value d). HACE is trained under both pairings, giving $3 \times 2 = 6$ HACE models per architecture–dataset pair. For SCE, the HACE-anchored pairing uses a rate $\ell \cdot d$ that depends on d, so three distinct SCE models are needed; the SCE-anchored pairing always uses ℓ, so a single SCE model covers all three dilution values. This yields $3 + 1 = 4$ SCE models per pair, for a total of 10 models per pair.

C Additional leaf-level results

C.1 Model-matched comparisons

Tables 8 and 9 compare each HACE model with the SCE model trained at the matching learning rate, for end-to-end training on FGVC and for linear probing on all datasets.

Table 8: Top-1 accuracy (%) of HACE and SCE on FGVC Aircraft (end-to-end), by architecture, dilution value d, and learning-rate pairing (standard / adj). Each HACE model is compared with the SCE model whose learning rate matches its leaf-level gradient magnitude (Appendix B.3). The better result per configuration is **bolded**.

	$d = 0.2$		$d = 0.2$ adj		$d = 0.5$		$d = 0.5$ adj		$d = 0.7$		$d = 0.7$ adj	
	HACE	SCE	HACE	SCE	HACE	SCE	HACE	SCE	HACE	SCE	HACE	SCE
ResNet-18	**73.16**	60.89	**78.88**	68.29	**70.93**	66.64	**69.87**	68.29	**69.40**	2.97	**70.78**	68.28
ResNet-34	71.92	**77.74**	**77.32**	70.33	**75.48**	2.13	**72.93**	70.33	**72.91**	5.01	**72.10**	70.33
ResNet-50	**71.56**	60.28	**80.62**	69.88	**69.28**	68.56	**76.03**	69.88	**70.90**	68.74	**70.15**	69.88
ConvNeXt-T	**33.21**	12.93	**52.24**	30.45	**30.78**	28.98	**38.82**	30.45	29.28	**30.15**	**36.66**	30.45
ViT-B/16	**18.48**	15.06	**15.84**	14.07	**17.04**	14.22	**20.67**	14.07	15.39	**15.72**	**16.17**	14.07
Swin-T	**41.67**	31.00	7.02	**38.16**	**39.75**	34.92	**44.34**	38.16	**40.41**	36.84	**40.71**	38.16

[2]HACE also receives gradient from the internal nodes of the hierarchy, which SCE does not; the adjustment is motivated only by the leaf-level contribution, which is directly comparable between the two losses.

Table 9: Top-1 accuracy (%) of HACE and SCE under linear probing on frozen DINOv2-Large features, by dataset, dilution value d, and learning-rate pairing (standard / adj). The better result per configuration is **bolded**.

	$d = 0.2$		$d = 0.2$ adj		$d = 0.5$		$d = 0.5$ adj		$d = 0.7$		$d = 0.7$ adj	
	HACE	SCE	HACE	SCE	HACE	SCE	HACE	SCE	HACE	SCE	HACE	SCE
CIFAR-100	**96.48**	95.53	**96.90**	96.62	**96.56**	96.53	**96.77**	96.62	**96.63**	96.52	**96.67**	96.62
FGVC	**41.36**	36.33	**58.97**	53.91	**50.27**	46.09	**57.58**	53.91	**52.71**	50.36	**56.56**	53.91
NABirds	**68.72**	68.65	**84.96**	83.49	**80.18**	78.56	**84.68**	83.49	80.18	**81.17**	**84.29**	83.49

C.2 Top-5 accuracy

Tables 10, 11 and 12 report top-5 accuracy for end-to-end training (best HACE vs. best SCE per pair), the baseline comparison on NABirds, and linear probing, following the format of the corresponding top-1 tables. In end-to-end training, HACE outperforms SCE in 17 of 18 architecture–dataset pairs, with a mean improvement of 4.46 pp. In linear probing, HACE achieves the highest and second-highest top-5 accuracy on all three datasets.

Table 10: Top-5 accuracy (%) of HACE and SCE in end-to-end training. Each cell reports the best result across dilution values and learning-rate pairings. The better result per pair is **bolded**.

	CIFAR-100		**FGVC**		**NABirds**	
	HACE	SCE	HACE	SCE	HACE	SCE
ResNet-18	**93.76**	93.01	**94.15**	88.63	**86.86**	83.73
ResNet-34	**94.87**	93.77	**93.97**	93.64	87.67	**88.08**
ResNet-50	**95.84**	94.63	**95.23**	90.58	**87.01**	80.73
ConvNeXt-T	**68.88**	67.17	**66.34**	57.01	**73.79**	53.42
ViT-B/16	**85.29**	85.25	**45.21**	33.99	**53.34**	51.23
Swin-T	**83.06**	80.04	**73.06**	66.31	**69.25**	66.16

Table 11: Top-5 accuracy (%) on NABirds under identical training conditions (same setting as Table 3). The best result per architecture is **bolded** and the second best underlined.

	HACE		**SCE**				**HXE**		**Margin**	
	No smooth.	**Label smooth.**	**No smooth.**	**Label smooth.**	**Soft label $\beta = 10$**	**Soft label $\beta = 30$**	$\alpha = 0.2$	$\alpha = 0.5$	$m = 2$	$m = 5$
ResNet-18	86.86	**87.70**	83.73	84.35	84.70	80.03	82.83	85.54	78.07	76.15
ResNet-34	87.67	**89.38**	88.08	88.93	88.09	82.73	85.53	88.64	80.32	79.40
ResNet-50	87.01	**89.65**	80.73	88.22	87.87	81.35	83.49	87.76	78.37	76.69
ConvNeXt-T	73.79	**79.04**	53.42	62.78	4.44	4.54	3.00	2.23	67.13	68.14
ViT-B/16	**53.34**	35.69	51.23	44.52	33.05	36.05	38.10	36.80	34.71	37.35
Swin-T	69.25	70.01	66.16	70.15	**71.90**	66.57	68.17	69.67	54.47	55.42

Table 12: Top-5 accuracy (%) under linear probing on frozen DINOv2-Large features. Each cell reports the best result across dilution values and learning-rate pairings. The best result per dataset is **bolded** and the second best underlined.

	HACE		**SCE**		**HXE**		**Soft Label**		**Margin**	
	No smooth.	**Label smooth.**	**No smooth.**	**Label smooth.**	$\alpha = 0.2$	$\alpha = 0.5$	$\beta = 10$	$\beta = 30$	$m = 2$	$m = 5$
CIFAR-100	99.03	**99.05**	98.94	98.97	98.96	98.92	98.99	98.96	99.00	98.94
FGVC	**86.02**	84.73	81.43	81.10	78.07	73.11	81.76	81.47	81.37	81.34
NABirds	**97.60**	**97.60**	97.21	97.11	96.02	88.35	97.23	97.21	79.12	96.69

D Additional hierarchical results

D.1 Per-class coarse-level accuracy

Figures 3–5 report per-class accuracy at the remaining coarse levels of each hierarchy (for NABirds we omit the species level, whose many classes cannot be displayed legibly), following the protocol

of Section 5.3 (ResNet-50, end-to-end). Internal-node predictions are obtained by aggregating probabilities upward; HACE produces these natively, while for SCE the aggregation is applied post-hoc. Grey bars show SCE accuracy and the colored segment the gain (teal) or decline (pink) with HACE; classes are sorted by SCE accuracy.

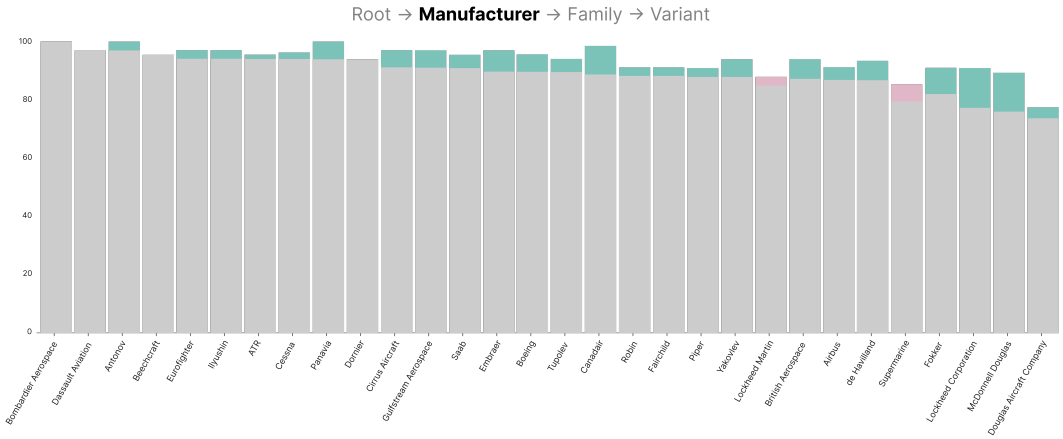

Figure 3: Per-class accuracy at the manufacturer level of FGVC Aircraft. HACE matches or outperforms SCE in 28 of 30 manufacturers (mean gain 4.21 pp).

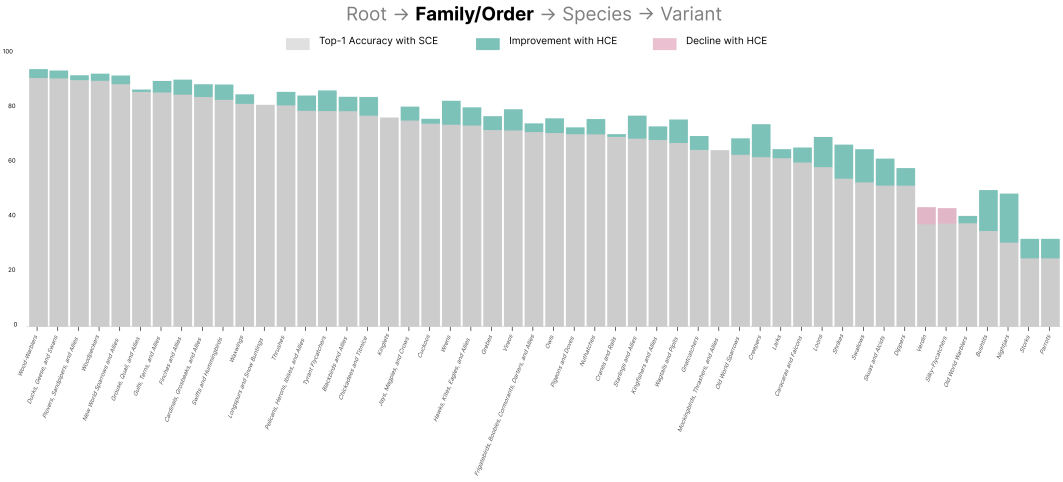

Figure 4: Per-class accuracy at the family/order level of NABirds, where perching birds (Passeriformes) are grouped by family and all other birds by order. HACE matches or outperforms SCE in 48 of 50 groups (mean gain 5.3 pp).

D.2 Operating-curve metrics per architecture

Table 13 reports the full set of hierarchical metrics on NABirds for every architecture and method variant; Table 4 in the main text summarizes it. F1 (Majority) is the information-based F1 of the prediction at threshold 0.5 (majority inference); R@95C is the exact recall at 95% correct.

Table 13: Hierarchical metrics (%) on NABirds (end-to-end). AP/AC: area under the information-precision vs. recall / correct vs. exact curves. R@90C/R@95C: exact recall at 90%/95% correct. F1 (Majority): information-based F1 at threshold-0.5 inference. F1 (Leaf): information-based F1 at top-1 leaf inference. Dist.: mean hierarchical distance at top-1 leaf inference (lower is better). HACE and SCE are reported without and with label smoothing (best-by-AP dilution and pairing); HXE and Margin are reported per hyperparameter. The best result per architecture and column is **bolded** and the second best underlined.

			From operating curves				Single operating point		
Model	**Loss**	**Variant**	**AP**	**AC**	**R@90C**	**R@95C**	**F1 (Majority)**	**F1 (Leaf)**	**Dist.**
ResNet-18	HACE	No smooth.	**76.25**	63.85	**65.95**	**58.70**	**72.23**	78.46	1.55
		Label smooth.	75.75	**64.51**	53.94	49.06	52.37	**80.18**	**1.42**
	SCE	No smooth.	70.98	59.57	46.41	42.23	42.63	76.62	1.66
		Label smooth.	70.61	59.06	45.76	41.54	41.64	76.33	1.68
	HXE	$\alpha = 0.2$	70.26	57.02	58.59	51.09	66.53	72.99	1.91
		$\alpha = 0.5$	74.01	61.33	62.25	55.03	69.21	76.63	1.67
	Margin	$m = 2$	65.75	52.53	53.91	46.02	65.30	68.61	2.20
		$m = 5$	64.66	51.86	52.69	44.82	66.37	67.57	2.27
ResNet-34	HACE	No smooth.	77.19	65.59	68.11	60.83	73.57	79.14	1.50
		Label smooth.	**79.38**	**68.97**	66.17	59.91	63.96	**82.07**	**1.30**
	SCE	No smooth.	76.55	65.86	60.79	55.31	55.80	79.89	1.44
		Label smooth.	78.50	67.83	64.61	58.07	59.73	81.38	1.34
	HXE	$\alpha = 0.2$	74.31	62.06	64.20	56.87	71.55	76.49	1.68
		$\alpha = 0.5$	78.56	67.04	**70.00**	**62.55**	**75.34**	80.43	1.41
	Margin	$m = 2$	68.53	55.80	57.31	49.66	68.10	71.13	2.03
		$m = 5$	68.73	56.51	57.69	49.43	70.15	71.28	2.02
ResNet-50	HACE	No smooth.	76.22	63.97	66.05	58.49	71.73	78.41	1.55
		Label smooth.	**79.35**	**69.04**	61.60	56.88	59.28	**82.75**	**1.25**
	SCE	No smooth.	76.37	65.79	56.33	52.01	51.23	80.46	1.40
		Label smooth.	76.63	65.96	55.89	51.71	51.03	80.67	1.39
	HXE	$\alpha = 0.2$	69.96	56.40	58.21	50.48	66.45	72.67	1.93
		$\alpha = 0.5$	76.84	64.60	**66.80**	**59.44**	**72.92**	79.02	1.51
	Margin	$m = 2$	66.18	52.58	53.65	45.66	65.46	69.16	2.17
		$m = 5$	65.16	51.77	52.62	44.55	66.67	68.16	2.23
ConvNeXt-T	HACE	No smooth.	61.35	46.80	**48.69**	**41.60**	**63.44**	64.48	2.49
		Label smooth.	**63.47**	**50.89**	38.08	34.67	40.43	**70.77**	**2.06**
	SCE	No smooth.	42.15	28.05	20.38	17.42	22.13	52.38	3.28
		Label smooth.	44.00	29.82	22.13	18.81	23.34	53.85	3.19
	HXE	$\alpha = 0.2$	7.68	0.01	3.56	2.28	9.35	7.81	6.13
		$\alpha = 0.5$	6.82	0.00	2.72	1.61	8.81	1.85	6.44
	Margin	$m = 2$	41.80	27.28	27.97	22.57	45.85	46.49	3.67
		$m = 5$	43.06	29.01	29.29	23.37	47.54	47.78	3.58
ViT-B/16	HACE	No smooth.	**36.94**	**21.37**	**22.86**	**17.87**	**36.69**	**42.55**	**3.94**
		Label smooth.	25.10	11.57	13.36	9.84	22.46	31.20	4.68
	SCE	No smooth.	21.44	8.70	10.65	7.46	17.64	27.75	4.90
		Label smooth.	30.92	16.33	17.33	13.13	27.02	37.17	4.30
	HXE	$\alpha = 0.2$	27.96	13.52	16.18	12.17	30.20	33.23	4.55
		$\alpha = 0.5$	26.75	12.42	15.16	11.31	29.50	32.11	4.62
	Margin	$m = 2$	25.81	12.73	14.22	10.40	29.30	30.92	4.69
		$m = 5$	27.39	14.60	15.16	11.19	31.84	32.85	4.57
Swin-T	HACE	No smooth.	54.58	38.77	**40.64**	33.49	57.36	58.52	2.89
		Label smooth.	49.11	34.73	20.33	17.90	22.86	59.99	2.78
	SCE	No smooth.	50.06	35.54	24.03	20.60	24.23	59.87	2.79
		Label smooth.	50.48	35.92	24.42	21.12	24.40	**60.20**	**2.77**
	HXE	$\alpha = 0.2$	53.26	37.46	39.11	32.44	56.08	57.32	2.97
		$\alpha = 0.5$	**54.83**	38.59	40.38	**34.14**	57.75	58.83	2.87
	Margin	$m = 2$	52.66	37.77	38.28	31.74	56.22	56.70	3.00
		$m = 5$	54.23	**39.76**	39.88	32.86	**58.02**	58.29	2.90

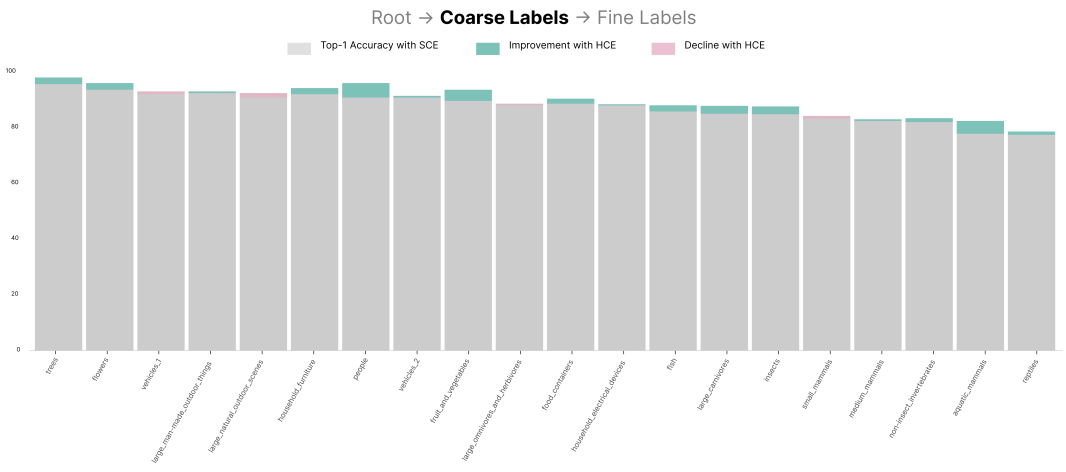

Figure 5: Per-class accuracy at the superclass level of CIFAR-100. HACE matches or outperforms SCE in 16 of 20 superclasses (mean gain 1.5 pp).