TREESCHISM: LOCALIZING AMBIGUITY IN HIERARCHIES

Anonymous authors
Paper under double-blind review

ABSTRACT

In classification, some inputs do not support fine-grained labels. For example, a blurry photo may show that a bird is a hawk, but not which type of hawk. A classifier presented with such an input must either guess its species (possibly communicating something false) or abstain (communicating nothing). A hierarchy of classes affords a third option: name the deepest node that the image justifies.

Existing hierarchical predictors either assume that ground-truth labels are leaves or can only reason about two-level taxonomies. We introduce TREESCHISM, which adapts evidential deep learning to localize ambiguity to a node of an arbitrarily deep taxonomy. A TREESCHISM model outputs scores at every node and accumulates them upward. This parameterization yields the *refinement gap*, the support for a node that does not apply to any of its descendants. We train the refinement gap to peak at the deepest label each sample warrants, and use it at inference to select a prediction granularity. Across seven taxonomies, TREESCHISM matches hyperevidential networks on the two-level taxonomies to which they are confined, improves exact-node accuracy over the strongest baseline by 6 to 21 points on deeper taxonomies, and achieves these successes without using held-out calibration data.

1 INTRODUCTION

Suppose a bird-watcher is asked to label images of birds with the species of their subjects. Figure 1 presents three images of flycatchers from the genus *Empidonax*, a group whose species are difficult to distinguish. The first two species can nevertheless be identified from their head shapes, bills, and wing structure. The third cannot. Although it is photographed just as clearly, it is either a Willow or an Alder Flycatcher, two species that cannot always be separated by sight alone (Lowther, 2020).

Which species should the bird-watcher record? They might make a guess (perhaps communicating incorrect information), or abstain from labelling entirely (communicating no information). Rather than commit to either extreme, the bird-watcher could instead exploit the taxonomy of species and label the specimen with its genus. In fact, the iNaturalist community has done exactly that: on that platform, the third image is labelled only with its genus, and not its species (iNaturalist contributors, 2026).

This example identifies a particular source of uncertainty in data, which we call *localized ambiguity*. Localized ambiguity arises when the label space is organized into a hierarchy, and a sample lacks sufficient detail to confidently label it with a leaf. Rather than guess or abstain, we *localize* the ambiguity to a non-leaf node of the taxonomy, labelling the sample with the deepest taxonomic node that can be assigned with confidence. This ambiguity is a property of the image and not the annotation process, so any successful classifier must defer to non-leaf labels when appropriate.

Most classification methods cannot express a non-leaf label: flat classifiers ignore the hierarchy entirely, while hierarchy-aware training losses (Valmadre, 2022; Bertinetto et al., 2020; Garg et al., 2022) assume leaf labels and only use a hierarchy to reduce mistake severity. Set-valued predictors (Angelopoulos et al., 2021) and partial-label learning (Wang et al., 2022) relax the single-label assumption, but reason over flat label sets rather than taxonomic depth.

Other work considers learning from samples whose labels are arbitrary nodes in a taxonomy (Brust et al., 2020; Chen et al., 2022; Park et al., 2026). These works observe that non-leaf labels arise for two different reasons: amiguity inherent to data, and uncertain annotators who defer to coarser labels

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.



(a) Hammond's Flycatcher (*E. hammondii*). *(b)* Western Flycatcher (*E. difficilis*). *(c)* An ambiguous specimen (*Empidonax* sp.).

Figure 1: Three flycatchers of the genus *Empidonax*. Photographs from iNaturalist (iNaturalist contributors, 2026), cropped: (a) © Nigel Voaden, CC BY, observation 2423854; (b) © Matt Felperin, CC BY, observation 200371722; (c) © Steven Bodzin, CC BY, observation 51173570.

rather than risk a mislabeling. Only one prior work offers a method to quantify inherent ambiguity and localize it to the appropriate node: Hyper-Evidential Neural Networks (HENNs) (Li et al., 2024), which quantify ambiguity (which they term *vagueness*[1]) as mass on a single fixed partition of the leaves. Thus, their method is restricted to two-level taxonomies, while we localize ambiguity in arbitrarily deep trees.

We propose TREESCHISM[2], a hierarchical classifier in which evidence for a node counts as evidence for all of its ancestors. Figure 2 compares TREESCHISM to flat classifiers and HENNs. Each node's evidence decomposes into two parts: an inherited part that is contributed by its descendants, and a local part that is not. Ambiguity is quantified as local evidence, e.g., as evidence for the genus *Empidonax* that no species accounts for. TREESCHISM can localize ambiguity to the appropriate node by measuring, at every node, the fraction of its evidence that is not inherited—its *refinement gap*. Our contributions are as follows.

1. We propose a parameterization in which each node accumulates evidence from its descendants, producing a refinement gap measurable at every node of a taxonomy.
2. We propose a training objective that drives the refinement gap to peak at the deepest justifiable label of each ambiguous training sample.
3. We propose an inference procedure that uses the refinement gap to select a prediction depth.
4. We empirically evaluate TREESCHISM's ability to localize ambiguity to appropriate nodes. We find that it matches HENNs (Li et al., 2024) on two-level taxonomies, and outperforms all other baselines on deeper taxonomies.

2 BACKGROUND

Hierarchical Label Spaces. Whereas traditional classifiers do not account for relationships between classes, hierarchical classification organizes them into a fixed *taxonomy* that encodes their relationships (Silla & Freitas, 2011). A taxonomy is a directed tree whose nodes are label classes and whose edges imply "is-a" relationships: if u is a descendant of v, then u is a sub-class of v. We can partition nodes into *taxonomic levels*, or levels, by the number of edges separating them from the root. The root is at level 0, the root's children at level 1, and so on. The root is a label that every sample satisfies, so it carries no information; we therefore only count non-root levels, calling a taxonomy with L of them an L-*level* taxonomy. *Higher* or *coarser* nodes are closer to the root, while *lower* or

[1] HENNs use *vagueness* in the Subjective Logic sense of the word; we use a different term to avoid conflation with that literature.

[2] TREESCHISM is named for the *schism* it predicts: the division along a root-to-leaf path where a node's support stops being inherited from its children, and a sample's ambiguity is localized.

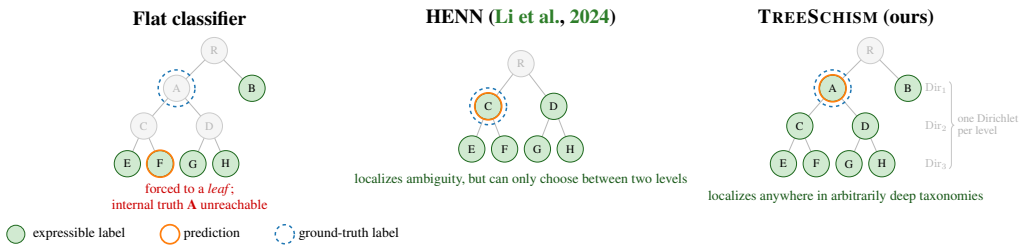

Figure 2: Green nodes are labels a method can emit; the orange ring marks its prediction for an ambiguous sample and the blue dashed ring that sample's ground-truth label. *Flat classifiers* (left) can only predict leaves. *HENN* (center) can predict internal nodes in two-level taxonomies. Only TREESCHISM (right) can identify the appropriate node for an ambiguous sample in an arbitrarily deep taxonomy.

finer nodes sit further away. Each taxonomy has one root and arbitrarily many leaves (nodes without children); nodes that are neither the root nor leaves are *internal nodes*.

Hierarchical Ambiguity. An emerging sub-field of hierarchical classification considers learning from instances with both leaf and non-leaf labels (Brust et al., 2020; Chen et al., 2022; Li et al., 2024; Park et al., 2026). These approaches assume that coarse (non-leaf) labels occur for two reasons: annotator inexpertise and ambiguity inherent to the data. If coarse labels arise from annotator inexpertise, a model should learn to resolve such instances to the appropriate fine-grained class. Conversely, if coarse labels arise from inherent ambiguity, a model should identify the ambiguity and select an appropriate prediction depth in the hierarchy. However, to the best of our knowledge, the sole existing method to do so is Hyper-Evidential Neural Networks (discussed below), which are confined to two-level taxonomies. Our method fills this gap by selecting appropriate prediction depths in arbitrarily deep taxonomies.

Evidential Deep Learning. Classical deep learning classifiers are trained to predict the parameters of a categorical distribution over classes, usually parameterized by softmax probabilities. Evidential Deep Learning (EDL) instead trains classifiers to predict the parameters of a Dirichlet distribution, the conjugate prior of the categorical (Sensoy et al., 2018). This approach allows a model to communicate its confidence in its predictions through two quantities: vacuity (uncertainty from a lack of evidence) and dissonance (uncertainty from conflicting evidence).

Hyper-Evidential Neural Networks. While EDL trains models to predict Dirichlet parameters, Hyper-Evidential Neural Networks train a model to predict hyper-Dirichlet parameters (Li et al., 2024). The ordinary Dirichlet distribution can only assign evidence to individual, mutually-exclusive classes, but the hyper-Dirichlet distribution can additionally assign evidence to arbitrary unions of those classes. This makes the hyper-Dirichlet a natural modelling choice for hierarchical label spaces. However, in its full generality, the hyper-Dirichlet lacks an analytic normalizer, which makes it difficult to deploy in practice. Li et al. (2024) propose to use a tractable restriction of the hyper-Dirichlet, called the grouped Dirichlet distribution (GDD). The GDD can only model two-level taxonomies, i.e., a single level of composite classes above the leaves.

HENNs are used to explicitly model and quantify inherent ambiguity in data (referred to as *vagueness* by Li et al. (2024)). If a HENN places evidence on singleton classes, it is stating that a sample is unambiguous and warrants a precise label. Conversely, evidence on composite classes indicates that a sample is ambiguous and cannot be resolved to a fine label.

3 METHOD

We propose TREESCHISM, a novel approach to hierarchical classification that explicitly models *localized ambiguity*. We say that an input's ambiguity is *localized* to a node in a fixed taxonomy when that node is the deepest (i.e., most precise) label that can be justified for the input. TREESCHISM is a method to train models capable of recognizing ambiguity and localizing it to appropriate nodes.

To achieve this, we adapt the Dirichlet parameterization of evidential deep learning (EDL) (Sensoy et al., 2018) to the hierarchical setting. While classical EDL models predict the parameters of a single Dirichlet distribution over a flat set of classes, TREESCHISM models predict the parameters of a separate Dirichlet distribution for each non-root taxonomic level. We ensure that predictions are coherent across levels by sharing parameters between ancestors and descendants.

3.1 THE DIRICHLET PARAMETERIZATION

In standard EDL (Sensoy et al., 2018), network outputs are non-negative *evidence vectors*, which are summed with a fixed *prior vector* to yield the parameters of a Dirichlet distribution. We modify this recipe in two ways: evidence is accumulated through a taxonomy before it is transformed to Dirichlet parameters, and the prior is chosen to reflect the taxonomy's structure.

TREESCHISM models output a real-valued score for every non-root taxonomic node. We apply the `softplus` activation function to convert these scores to non-negative *local evidence* scores. Let $e_n^{(\text{local})}$ denote node n's local evidence. This evidence is local because, as we explain below, it is the evidence for node n not accounted for by any of its descendants.

To enforce hierarchical relationships, every node shares its local evidence with its ancestors, yielding the *accumulated evidence* scores $e_n^{(\text{acc})}$. Hierarchies encode "is-a" relationships between ancestors and descendants. Thus, evidence in support of any node is necessarily also evidence in support of its ancestors. Leaves have no descendants, so their local and accumulated evidence scores coincide: $e_n^{(\text{acc})} = e_n^{(\text{local})}$ for every leaf n. At any other non-root node, accumulated evidence is defined by the recurrence

$$e_n^{(\text{acc})} \;=\; e_n^{(\text{local})} \;+\; \sum_{c \text{ child of } n} e_c^{(\text{acc})}. \tag{1}$$

Thus, a node's accumulated evidence is the sum of all local evidence in its subtree. Local evidence lets us model hierarchical ambiguity naturally. It can identify a sample as, e.g., a mammal, without committing to any mammal in particular.

We once again adopt an EDL-inspired approach in adding a fixed prior to evidence to obtain Dirichlet parameters. We fix a uniform prior weight over leaf classes, and set each internal node's prior weight to the sum of its children's priors. The prior weights are negligible by design (0.01 on leaves), ensuring that Dirichlet parameters are positive and hence non-degenerate without providing significant support for any class. If W_n is the prior at node n, then n's Dirichlet parameter is $\alpha_n = e_n^{(\text{acc})} + W_n$. Every parent's prior is the sum of its children's to ensure that local evidence is the sole source of disagreement between adjacent taxonomic levels. By collecting the Dirichlet parameters at each level ℓ into a vector $\alpha^{(\ell)}$, we obtain a separate Dirichlet distribution $\mathrm{Dir}\big(\alpha^{(\ell)}\big)$ over every level.

3.2 TRAINING OBJECTIVE

Suppose a training sample is labelled with node t, indicating that t and its ancestors are the correct labels for the sample. We call t the *supervision boundary*. It can be a leaf or an internal node. Training TREESCHISM has two goals. First, at every level along the chain from root to t, the model should identify the correct node. That is, each such level's Dirichlet $\mathrm{Dir}\big(\alpha^{(\ell)}\big)$ should concentrate evidence on the supervised ancestor and withhold it from other nodes. We enforce this with an EDL-style loss, applied level-wise. Second, the model should learn where the supervision boundary is, allowing it to localize ambiguity to the appropriate node. As we discussed in §3.1, local evidence can express localized ambiguity: a node's local evidence is large precisely when its support is not inherited from its children. Thus, we want local evidence to be large at the supervision boundary, whether it be a leaf or an internal node, and small everywhere else.

The refinement gap. To make our second goal trainable, we need a measure of how much of a node's evidence is local. We define the *refinement gap* at a node n as

$$r_n \;:=\; \frac{e_n^{(\text{local})}}{e_n^{(\text{acc})} + W_n} \;=\; \frac{e_n^{(\text{local})}}{\alpha_n} \;\in\; [0, 1), \tag{2}$$

the fraction of n's Dirichlet parameter α_n contributed by its own local evidence rather than its descendants or the prior. The refinement gap is near zero when a node inherits all its evidence from

its descendants, and close to one when most of its evidence is local. Thus, a large refinement gap can identify the supervision boundary.

The training loss. Let t be the supervision boundary. For a node n, let S_n denote the sum of the Dirichlet parameters at n's level; moreover, let $\mathrm{anc}(t)$ denote the strict ancestors of t, excluding the root. The loss is then

$$\mathcal{L}(t) = \underbrace{\sum_{p \in \mathrm{anc}(t) \cup \{t\}} \left[\log S_p - \log \alpha_p\right]}_{\text{(i) per-level EDL classification}} + \lambda \underbrace{\left[\mathbf{1}[t \text{ internal}]\left(-\log r_t\right) + \sum_{p \in \mathrm{anc}(t)} \left(-\log(1 - r_p)\right)\right]}_{\text{(ii) refinement-gap supervision}}. \tag{3}$$

Term (i) is the standard EDL log-loss (Sensoy et al., 2018) applied at each supervised level. It is the negative logarithm of the Dirichlet's expected probability of the supervised ancestor, and drives evidence onto the correct node. This term is only computed over supervised levels, but evidence from other levels still enters via accumulation (§3.1). Term (ii) supervises the refinement gap with binary cross-entropy. If the supervision boundary t is internal, $-\log r_t$ drives r_t to 1, teaching the model to produce local evidence at the supervision boundary. At each strict ancestor p, $-\log(1 - r_p)$ drives r_p toward 0, teaching the model to rely on evidence accumulated from t rather than produce further local evidence higher up the tree. The positive scalar λ is a tunable hyper-parameter.

The two terms of the loss may appear incompatible. For a strict ancestor n of the supervision boundary t, the first term wants local evidence at n while the second penalizes it. However, this loss can reliably train useful models (§4). We explain its empirical effectiveness by studying its gradients in §E.

3.3 Inference

At inference time, TreeSchism uses the refinement gap to localize ambiguity to the appropriate node. Inference starts at the root node and walks down the taxonomy, moving from a parent to its most-likely child, as measured by the Dirichlet over the child level's nodes. At each node n, we compute the refinement gap r_n. The first time we measure $r_n > \tau$ for a fixed threshold τ, we halt inference and predict n. If we reach a leaf n, we output it as our prediction. Since the refinement gap lies in $[0, 1)$, the threshold τ has a consistent, interpretable scale across taxonomic levels. Rather than tune it, we take $\tau = 0.5$ in all of our experiments; §4.4 empirically justifies this choice.

4 Experiments

4.1 Experimental Setup

Datasets. Our goal in this section is to empirically demonstrate that TreeSchism can localize data-inherent ambiguity to specific nodes in arbitrarily deep taxonomies. Li et al. (2024) evaluate the ability of HENNs (§2) to identify ambiguity in the two-level taxonomies associated with the BREEDS (Santurkar et al., 2021) and TinyImageNet (Fei-Fei et al., 2015) datasets. They synthetically generate ambiguous samples by applying Gaussian blurring to ground-truth samples, and train a model to classify such samples at the taxonomy's interior level. To compare TreeSchism against HENNs directly, we use the same blurring mechanism to evaluate on three datasets: the BREEDS datasets living17 and nonliving26, as well as TinyImageNet.

The primary innovation of TreeSchism over HENNs is its ability to localize ambiguity to nodes in arbitrarily deep taxonomies. To evaluate its performance on this task, we consider the BIOSCAN-5M dataset (Gharaee et al., 2024). This dataset consists of insects labelled at varying depths of a biological taxonomy (e.g., species, genus, family). The data were collected as part of an international effort to measure and promote insect biodiversity. While the full dataset is multi-modal, consisting of images, geographic locations, DNA sequences, and more, we use only the images for the purposes of training and evaluation. We consider four depth-four subtrees of the full BIOSCAN-5M dataset separately, each corresponding to one insect order: Hemiptera, Coleoptera, Hymenoptera, and Lepidoptera. Each order contains visually distinct insects (e.g., moths vs. beetles), and they differ in the shape of their taxonomies as well as in the distribution of ambiguous samples across taxonomic levels. More details are given in Appendix D.

Table 1: **Set-prediction quality.** OverJS / CompJS (%, test), mean ± std dev over three seeds; higher is better. We identify nodes with the sets of their leaf descendants to compare hierarchical and set-valued predictors. Datasets are listed with their taxonomies' depths in parentheses. **Bold**: best in column, or within one std dev of it. HENN can only be used with the three two-level taxonomies. All baselines except HENN are calibrated on held-out data; TREESCHISM requires no calibration.

	Lepidoptera (4)		Coleoptera (4)		Hymenoptera (4)		Hemiptera (4)		TinyImageNet (2)		NonLiving26 (2)		Living17 (2)	
Method	OverJS	CompJS	OverJS	CompJS	OverJS	CompJS	OverJS	CompJS	OverJS	CompJS	OverJS	CompJS	OverJS	CompJS
HENN	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	82.8 ±0.0	**89.2 ±0.3**	**87.5 ±0.3**	**92.6 ±0.2**	88.1 ±0.1	**95.3 ±0.1**
SoftCE+HSC	**68.6 ±0.1**	**76.6 ±0.2**	**86.0 ±0.2**	**89.9 ±0.2**	**78.6 ±0.2**	**84.8 ±0.2**	**84.2 ±0.1**	**90.3 ±0.1**	80.1 ±0.1	60.9 ±9.2	84.4 ±0.6	90.6 ±0.4	86.5 ±0.2	94.4 ±0.1
SoftCE+THR	27.6 ±0.1	18.7 ±0.1	42.3 ±0.0	38.5 ±0.0	31.8 ±0.1	26.0 ±0.1	38.8 ±0.1	31.6 ±0.1	80.9 ±0.3	78.8 ±1.1	84.4 ±0.2	91.0 ±1.5	86.2 ±0.3	93.3 ±0.1
SoftCE+RAPS	20.1 ±0.1	7.5 ±0.0	20.2 ±0.1	12.3 ±0.0	17.1 ±0.1	7.0 ±0.0	22.9 ±0.1	10.5 ±0.0	80.0 ±0.1	55.4 ±0.4	54.9 ±0.0	23.5 ±0.0	57.2 ±0.1	24.4 ±0.0
CHILLAX+HSC	39.0 ±0.5	41.0 ±0.9	61.0 ±0.7	63.2 ±1.8	53.5 ±0.5	56.8 ±0.5	53.8 ±1.2	56.9 ±1.2	80.0 ±0.2	55.3 ±0.6	55.7 ±0.1	27.8 ±3.1	61.3 ±0.1	52.6 ±0.3
CHILLAX+THR	21.8 ±0.1	9.0 ±0.1	26.9 ±0.0	19.6 ±0.0	19.6 ±0.0	9.1 ±0.0	24.8 ±0.1	12.3 ±0.1	80.0 ±0.2	55.4 ±0.6	55.9 ±0.1	24.9 ±0.4	58.9 ±0.0	29.8 ±0.8
HRN+HSC	25.9 ±1.2	21.4 ±1.7	52.0 ±1.9	50.0 ±2.0	59.8 ±1.2	58.3 ±1.6	56.6 ±2.2	53.8 ±2.5	79.0 ±0.2	51.5 ±0.7	53.5 ±0.5	46.3 ±1.3	61.0 ±0.4	55.4 ±0.2
HRN+THR	15.3 ±0.2	4.8 ±0.1	15.6 ±0.3	8.1 ±0.4	17.9 ±0.3	8.7 ±0.3	21.3 ±0.3	9.5 ±0.3	78.9 ±0.2	51.8 ±0.8	55.2 ±0.1	23.5 ±0.1	57.6 ±0.2	24.9 ±0.3
TREESCHISM (ours)	**69.6 ±1.1**	75.8 ±0.8	85.8 ±0.1	**89.8 ±0.0**	77.7 ±0.2	83.1 ±0.0	**84.3 ±0.3**	89.1 ±0.4	**83.1 ±0.0**	88.8 ±0.5	**87.6 ±0.2**	**92.6 ±0.1**	**88.3 ±0.2**	94.5 ±0.1

Baselines. Although a line of research modelling hierarchical ambiguity has been developed over the course of several years, to the best of our knowledge, HENNs (Li et al., 2024) are the only prior method that offers a principled mechanism to select a taxonomic level. To evaluate TREESCHISM against certain baselines, we must therefore prescribe our own inference procedures on top of them.

The four baseline models we consider are as follows. The first is **HENN**, which we described in §2. **SoftCE** trains an ordinary classifier using a cross-entropy loss. A leaf-labelled sample uses a standard one-hot target, while internal-node-labelled targets are given soft targets that spread mass uniformly over the label node's leaf descendants. **CHILLAX** (Brust et al., 2020) and Hierarchical Residual Networks (**HRN**; Chen et al., 2022) model hierarchical ambiguity explicitly (§2). Given a sample labelled with an internal node, CHILLAX's masked loss leaves all nodes below the label unsupervised, while HRN uses an energy score over all leaf descendants simultaneously.

Inference Procedures. The inference procedures used for SoftCE, CHILLAX, and HRN are as follows. The first is **THR** ("threshold"), which predicts sets of leaves by thresholding per-leaf probabilities. Each model's native output is converted into normalized probabilities over the leaves alone, and a leaf is included in the prediction set if and only if its probability exceeds a threshold τ; when no leaves clear τ, the single most probable leaf is included. The second is **HSC** (Hierarchical Selective Classification), a procedure to use softmax probabilities over leaves to predict nodes. Practitioners specify a target mis-coverage rate α and tune an inference procedure on held-out data to achieve it. For SoftCE models alone, we also use **RAPS**, a conformal-prediction inference procedure (Angelopoulos et al., 2021). It predicts sets of leaves. Like HSC, it requires held-out calibration data to calibrate its procedure to achieve a desired mis-coverage level. Each of these three inference procedures require that we fix an operating point (e.g., target coverage level). We select operating points from a fixed grid (§B) to maximize CompJS (defined below in §4.2) over a held-out calibration set.

Reproducibility. See §B for a complete description of our experimental setup.

4.2 SET-PREDICTION QUALITY

Our inference procedures fall into two classes: set predictors (THR, RAPS), which output sets of leaves; and hierarchical predictors (HSC, HENN, TREESCHISM), which output a single taxonomic node. To compare these on data labelled with taxonomic nodes, we follow Li et al. (2024) in identifying every node with the set of its leaf descendants. Thus, every label and prediction is interpreted as a set of leaves, and we score a prediction using its Jaccard similarity to the true set. Given two finite sets A and B, not both empty, their Jaccard similarity is $\mathrm{JS}(A, B) = |A \cap B|/|A \cup B|$. Again following Li et al. (2024), we let **OverJS** be the mean Jaccard similarity score across *all* test samples, while **CompJS** is the same metric computed only for samples with non-leaf labels.

We present our measurements in Table 1. TREESCHISM, SoftCE+HSC, and HENN outperform other approaches by wide margins. **The hierarchical predictors consistently beat set predictors**. Interestingly, although HRN and CHILLAX are designed to model hierarchical ambiguity, HSC inference performs much better on SoftCE.

4.3 Localization Quality

Having established the superiority of hierarchical predictors for this task, we now report hierarchy-specific metrics to compare hierarchical predictors. The metrics we report are specialized to our setting, and are defined in the caption of Table 2, where we report the results. TreeSchism has the best ExactNode% and Level-Δ on six of seven datasets, competitive with HENN on shallow taxonomies and winning the others. Conversely, HSC procedures yield the lowest, hence best, OfLineage% scores. All methods except HRN are virtually tied on Acc@Level%, so the other methods are primarily separated by how well they choose the prediction depth. HRN's underperformance is concentrated on the four deep BIOSCAN-5M datasets, indicating that its auxiliary leaf-level loss transfers poorly when supervision frequently terminates above the leaves.

We now offer further comparison of TreeSchism and SoftCE+HSC, its closest competitor on the ExactNode% and Level-Δ metrics. Both methods have similar inference procedures: they select a branch from root to leaf, and walk it until a node's score first exceeds some acceptance threshold. The methods use different scores: TreeSchism uses the refinement gap (§3.3) and SoftCE+HSC uses the summed softmax probabilities of a node's leaf descendants. Table 3 reports AUROC scores measuring how well a stop signal predicts whether a given node is the correct stopping point for a sample. These inference procedures walk the tree in opposite directions, but note that since SoftCE+HSC's accumulated scores increase from leaf to root, its ascending inference procedure admits an equivalent formulation as a descending inference procedure. Thus, TreeSchism and SoftCE+HSC's inference procedures differ in signal but not direction, and our present comparison is fair to both.

For both methods, we measure two stop signals' prediction strengths using the AUROC in Table 3. The *refinement gap* is TreeSchism's native score, and *marginal P(child)* is SoftCE+HSC's. We also evaluate both methods using the *child-conditional max* score, defined as the share of a node's probability mass taken by its most-likely child. This is the probability-space analogue of the refinement gap. We find that **the refinement gap is always the strongest predictor.** Moreover, when using child-conditional max scoring, SoftCE+HSC outperforms TreeSchism on average. Since TreeSchism and SoftCE are equally capable of identifying the true node within its level (Acc@Level%, Table 2), TreeSchism's lead can be solely attributed to its use of the refinement gap, which measures ambiguity rather than model confidence.

4.4 Operating Curves

At inference time, TreeSchism predicts the first node whose refinement gap exceeds 0.5. HSC inference also has a threshold that determines when it commits to a node, but that threshold is not fixed *a priori*. Figure 3 plots OverJS against operating points for both TreeSchism and SoftCE+HSC. TreeSchism's curves are remarkably flat, and $\tau = 0.5$ sits near the peak across all seven datasets. Conversely, SoftCE+HSC's curves peak at operating points ranging from $\alpha \approx 0.75$ to $\alpha \approx 0.95$, and OverJS often falls off a cliff near the optimal α. Thus, unlike the baselines, **TreeSchism can make good predictions without calibrating its decision rule**.

5 Related Work

Hierarchical Ambiguity. CHILLAX (Brust et al., 2020) and Hierarchical Residual Networks (Chen et al., 2022) both learn from non-leaf labels, but treat them as deficient supervision to see through rather than as ambiguity to predict. Neither supplies a rule for choosing how deep to predict; in §4, we furnish them with inference procedures and demonstrate that they are less capable of quantifying ambiguity than TreeSchism. Park et al. (2026) cope with non-leaf labels by either sourcing additional supervision from a VLM or adopting a self-supervised learning approach. Both of these approaches teach a model to predict finer labels than were supplied by the annotation process, whereas our method is intended for cases where the training data's annotations are trusted, such as the iNaturalist example of Figure 1.

Selective prediction and deferral. Selective classification lets a model opt *not* to predict (Chow, 1970; El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017). Classical methods offer a single "reject" option, which communicates no information about the instance under consideration; TreeSchism can defer to any of the distinct taxonomic granularities available to it. Learning to defer (Madras et al.,

Table 2: **Localization quality** of hierarchical inference procedures (test, mean ± std over three seeds). **Bold**: best in column, or within one std of it. ExactNode%: committed node is exactly the true node (higher is better). Level-Δ: mean |predicted − true depth| (lower is better). OffLineage%: fraction of predictions that are neither ancestors nor descendants of the true node (lower is better). Acc@Level%: percentage of samples whose true class is ranked most likely *at their true level*, ignoring the method's predicted depth; higher is better.

Method	Lepidoptera	Coleoptera	Hymenoptera	Hemiptera	TinyImageNet	NonLiving26	Living17
ExactNode% (↑)							
HENN	N/A	N/A	N/A	N/A	**82.7 ±0.0**	**86.8 ±0.3**	86.5 ±0.2
SoftCE+HSC	53.4 ±0.2	61.4 ±0.5	66.0 ±0.1	76.5 ±0.2	76.1 ±0.3	82.5 ±0.8	84.0 ±0.2
CHILLAX+HSC	26.1 ±0.5	36.2 ±0.6	32.4 ±0.2	35.6 ±1.2	75.9 ±0.3	44.4 ±0.1	50.8 ±0.1
HRN+HSC	18.0 ±1.1	37.6 ±1.3	44.6 ±1.3	45.1 ±1.1	75.2 ±0.2	46.8 ±0.2	53.2 ±0.4
TREESCHISM (ours)	**67.9 ±1.2**	**82.7 ±0.1**	**73.7 ±0.2**	**82.6 ±0.3**	82.4 ±0.1	**86.9 ±0.3**	**86.9 ±0.2**
Level-Δ (↓)							
HENN	N/A	N/A	N/A	N/A	**0.00 ±0.00**	**0.03 ±0.00**	0.06 ±0.00
SoftCE+HSC	0.72 ±0.00	0.51 ±0.01	0.48 ±0.00	0.36 ±0.00	0.10 ±0.05	0.14 ±0.01	0.12 ±0.00
CHILLAX+HSC	1.33 ±0.01	1.02 ±0.03	1.00 ±0.01	0.97 ±0.01	0.07 ±0.00	0.53 ±0.03	0.52 ±0.00
HRN+HSC	1.49 ±0.03	0.97 ±0.02	0.77 ±0.03	0.79 ±0.06	0.07 ±0.00	0.64 ±0.01	0.53 ±0.01
TREESCHISM (ours)	**0.43 ±0.02**	**0.22 ±0.00**	**0.36 ±0.01**	**0.26 ±0.01**	0.01 ±0.00	**0.03 ±0.00**	**0.06 ±0.00**
OffLineage% (↓)							
HENN	N/A	N/A	N/A	N/A	**17.2 ±0.0**	10.3 ±0.1	7.2 ±0.2
SoftCE+HSC	9.1 ±0.1	2.1 ±0.1	2.3 ±0.0	2.2 ±0.0	**16.1 ±2.4**	6.0 ±0.4	4.9 ±0.1
CHILLAX+HSC	**2.1 ±0.3**	**0.8 ±0.1**	**1.2 ±0.0**	**0.9 ±0.0**	**17.5 ±0.3**	6.7 ±2.0	**1.3 ±0.0**
HRN+HSC	6.4 ±0.6	6.3 ±0.3	5.4 ±0.4	7.5 ±0.5	18.7 ±0.2	**1.3 ±0.1**	1.7 ±0.1
TREESCHISM (ours)	19.6 ±0.5	5.2 ±0.1	7.2 ±0.2	6.1 ±0.2	**16.8 ±0.0**	10.1 ±0.2	7.5 ±0.1
Acc@Level% (↑)							
HENN	N/A	N/A	N/A	N/A	82.7 ±0.1	89.2 ±0.1	91.7 ±0.2
SoftCE+HSC	79.0 ±0.2	93.6 ±0.0	89.5 ±0.1	91.6 ±0.0	82.5 ±0.1	89.5 ±0.0	91.7 ±0.1
CHILLAX+HSC	**80.1 ±0.4**	**95.9 ±0.1**	**91.0 ±0.1**	**92.3 ±0.2**	82.5 ±0.2	**90.0 ±0.0**	**92.4 ±0.0**
HRN+HSC	70.1 ±0.9	90.7 ±1.0	87.0 ±0.2	90.2 ±0.1	81.5 ±0.2	89.5 ±0.2	91.7 ±0.2
TREESCHISM (ours)	78.3 ±0.6	93.3 ±0.0	90.0 ±0.1	91.9 ±0.1	**83.2 ±0.0**	89.5 ±0.2	91.8 ±0.1

Table 3: **Stop-signal quality** (test, mean ± std over three seeds). Each dataset's taxonomy's depth is listed alongside it in parentheses. **Bold**: best in column within a level block, or within one std of it; higher is better. The *refinement gap* and *marginal P(child)* are the native ambiguity-detection scores for TREESCHISM and SoftCE+HSC, respectively; *child-conditional max* is a probability-space analogue of the refinement gap.

Signal	Model	Lepidoptera (4)	Coleoptera (4)	Hymenoptera (4)	Hemiptera (4)	TinyImageNet (2)	NonLiving26 (2)	Living17 (2)
Stop-signal AUROC, level 1 (↑)								
marginal $P(\text{child})$	SoftCE	0.699 ±0.002	0.763 ±0.003	0.798 ±0.003	0.861 ±0.004	0.653 ±0.006	0.929 ±0.003	0.950 ±0.002
child-conditional max	SoftCE	0.759 ±0.001	0.749 ±0.001	0.781 ±0.001	0.858 ±0.002	0.662 ±0.001	0.986 ±0.002	0.972 ±0.001
refinement gap r_n	TREESCHISM	**0.913** ±0.004	**0.957** ±0.001	**0.929** ±0.001	**0.956** ±0.002	**0.995** ±0.000	**0.992** ±0.001	**0.986** ±0.000
child-conditional max	TREESCHISM	0.702 ±0.007	0.673 ±0.004	0.639 ±0.010	0.838 ±0.008	0.629 ±0.004	0.953 ±0.013	0.920 ±0.005
Stop-signal AUROC, level 2 (↑)								
marginal $P(\text{child})$	SoftCE	0.726 ±0.001	0.840 ±0.002	0.816 ±0.001	0.830 ±0.001	N/A	N/A	N/A
child-conditional max	SoftCE	0.786 ±0.002	0.878 ±0.002	0.848 ±0.002	0.859 ±0.002	N/A	N/A	N/A
refinement gap r_n	TREESCHISM	**0.861** ±0.003	**0.973** ±0.001	**0.966** ±0.001	**0.955** ±0.002	N/A	N/A	N/A
child-conditional max	TREESCHISM	0.701 ±0.011	0.890 ±0.003	0.824 ±0.012	0.818 ±0.008	N/A	N/A	N/A
Stop-signal AUROC, level 3 (↑)								
marginal $P(\text{child})$	SoftCE	0.640 ±0.001	0.669 ±0.004	0.701 ±0.001	0.670 ±0.002	N/A	N/A	N/A
child-conditional max	SoftCE	0.746 ±0.000	0.727 ±0.000	0.755 ±0.001	0.805 ±0.002	N/A	N/A	N/A
refinement gap r_n	TREESCHISM	**0.884** ±0.002	**0.928** ±0.001	**0.875** ±0.002	**0.934** ±0.002	N/A	N/A	N/A
child-conditional max	TREESCHISM	0.717 ±0.007	0.692 ±0.001	0.686 ±0.008	0.779 ±0.002	N/A	N/A	N/A

2018; Mozannar & Sontag, 2020) routes hard samples to a human expert; we train a model to quantify the degree of ambiguity in such samples. Conformal prediction methods offer distribution-free coverage guarantees, but in practice require held-out calibration data and return flat leaf sets rather than taxonomic nodes (Angelopoulos & Bates, 2022). Hierarchical Selective Classification (Goren et al., 2024) predicts arbitrary taxonomic nodes rather than leaves, but requires held-out calibration data and does not measure ambiguity directly.

6 DISCUSSION & LIMITATIONS

Ambiguity is not confidence. TREESCHISM's advantage comes from *what* it measures, rather than from a stronger classifier. Table 2 shows that TREESCHISM and SoftCE identify the correct node within a level equally well (Acc@Level%); they differ only in what signal is used to select the prediction depth. In Table 3, we see that under the child-conditional max score (the probability-space analogue of the refinement gap), SoftCE outperforms TREESCHISM. SoftCE+HSC derives

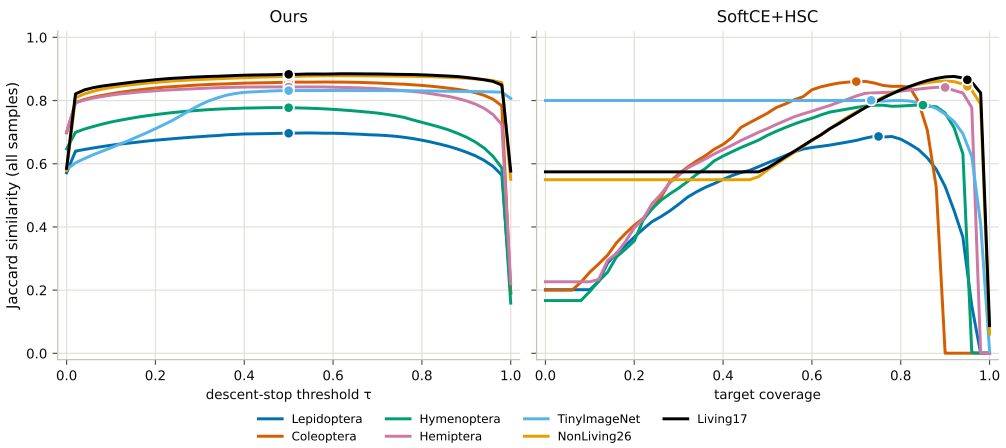

Figure 3: **Operating curves** (mean over three seeds; dots are deployed operating points). Left: TREESCHISM swept over its descent-stop threshold τ (§3.3); right: SoftCE+HSC swept over its target mis-coverage rate α. HSC's OverJS is highly sensitive to α, and the optimal α varies across datasets; this method requires extra held-out calibration data. Conversely, TREESCHISM's OverJS is insensitive to its operating point.

internal-node scores from leaf softmax probabilities, and hence it cannot model disagreement between ancestors and descendants. TREESCHISM's ability to do so allows it to more accurately identify the true prediction depth.

Limitations. Three of our seven datasets derive their ambiguity from Gaussian blurring (Li et al., 2024), which is a proxy for naturally-occurring ambiguity in deep taxonomies. Although TREESCHISM can make good decisions without calibrating its decision rule (§4.4), this comes at the cost of a training hyperparameter λ that requires tuning (§3.2). TREESCHISM assumes that non-leaf labels are solely the result of ambiguity inherent to the data, and never a consequence of the annotation process. Although similar assumptions are made throughout the literature, to the best of our knowledge, no prior work discusses how to test whether any such assumption holds on real-world data, an important direction for future work. Finally, the $\tau = 0.5$ threshold used during inference (§3.3) is supported empirically rather than derived, and taxonomies differing sharply in depth or branching factor might shift its optimum.

Future work. The refinement gap is a per-node score, so a system that routes a sample to a human expert at the node where TREESCHISM halts would compose our method with learning to defer (Madras et al., 2018; Mozannar & Sontag, 2020), replacing a binary routing decision with a graded one. Separately, the accumulation recurrence (1) assumes that each node has a single parent, so TREESCHISM does not yet apply to label spaces structured as directed acyclic graphs rather than trees. Extending it would broaden the settings in which localized ambiguity can be modelled.

7 CONCLUSION

When an input cannot justify a leaf label, a hierarchical classifier can name a non-leaf node rather than guess or abstain. We formalized this as *localized ambiguity*, defined at every node of a taxonomy, and introduced TREESCHISM, which adapts evidential deep learning to arbitrarily deep taxonomies by accumulating local evidence through a hierarchy. The *refinement gap* that this parameterization yields measures how much of a node's support is its own rather than inherited from its children, turning the choice of prediction depth into a calibration-free thresholding decision. Across seven taxonomies, TREESCHISM matches methods confined to two levels and localizes ambiguity more accurately than every baseline on deeper ones.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2022. URL https://arxiv.org/abs/2107.07511. 8

Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=eNdiU_DbM9. 1, 6, 12

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12503–12512, 2020. doi:10.1109/CVPR42600.2020.01252. 1

Clemens-Alexander Brust, Björn Barz, and Joachim Denzler. Making every label count: Handling semantic imprecision by integrating domain knowledge, 2020. URL https://arxiv.org/abs/2010.06469. 1, 3, 6, 7

Jingzhou Chen, Peng Wang, Jian Liu, and Yuntao Qian. Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification, 2022. URL https://arxiv.org/abs/2201.03194. 1, 3, 6, 7

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory*, 16(1):41–46, 1970. doi:10.1109/TIT.1970.1054406. URL https://doi.org/10.1109/TIT.1970.1054406. 7

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010. URL http://jmlr.org/papers/v11/el-yaniv10a.html. 7

Li Fei-Fei, Andrej Karpathy, and Justin Johnson. Tiny imagenet visual recognition challenge, 2015. URL https://tiny-imagenet.herokuapp.com. 5

Ashima Garg, Depanshu Sani, and Saket Anand. Learning hierarchy aware features for reducing mistake severity. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pp. 252–267, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-20052-6. doi:10.1007/978-3-031-20053-3_15. URL https://doi.org/10.1007/978-3-031-20053-3_15. 1

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4885–4894, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. 7

Zahra Gharaee, Scott C. Lowe, ZeMing Gong, Pablo Millan Arias, Nicholas Pellegrino, Austin T. Wang, Joakim Bruslund Haurum, Iuliia Zarubiieva, Lila Kari, Dirk Steinke, Graham W. Taylor, Paul Fieguth, and Angel X. Chang. BIOSCAN-5M: A multimodal dataset for insect biodiversity. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 36285–36313. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/3fdbb472813041c9ecef04c20c2b1e5a-Paper-Datasets_and_Benchmarks_Track.pdf. 5

Shani Goren, Ido Galil, and Ran El-Yaniv. Hierarchical selective classification. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wzof7Y66xs. 8, 12

iNaturalist contributors. iNaturalist research-grade observations. Occurrence dataset, 2026. 1, 2

Changbin Li, Kangshuo Li, Yuzhe Ou, Lance M. Kaplan, Audun Jøsang, Jin-Hee Cho, DONG HYUN JEONG, and Feng Chen. Hyper evidential deep learning to quantify composite classification uncertainty. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=A7t7z6g6tM. 2, 3, 5, 6, 9, 12

P. E. Lowther. *Alder Flycatcher (Empidonax alnorum)*. Cornell Lab of Ornithology, Ithaca, NY, USA, version 1.0 edition, 2020. doi:10.2173/bow.aldfly.01. URL https://doi.org/10.2173/bow.aldfly.01. 1

David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/09d37c08f7b129e96277388757530c72-Paper.pdf. 7, 9

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html. 8, 9

Seulki Park, Zilin Wang, and Stella X. Yu. Free-grained hierarchical visual recognition, 2026. URL https://arxiv.org/abs/2510.14737. 1, 3, 7

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. In *International Conference on Learning Representations*, 2021. 5

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3183–3193, Red Hook, NY, USA, 2018. Curran Associates Inc. 3, 4, 5

Carlos N. Silla and Alex A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2011. ISSN 1573-756X. doi:10.1007/s10618-010-0175-9. URL https://doi.org/10.1007/s10618-010-0175-9. 2

Jack Valmadre. Hierarchical classification at multiple operating points. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=mNtFhoNRr4i. 1

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=EhYjZy6e1gJ. 1

A AI Disclosure

In this work, we used generative AI tools to help implement both our method and the baselines in code, to identify instances of incorrect spelling/grammar and unclear or awkward wording in our manuscript, for assistance in the gradient analysis given in §E, and for feedback and criticism of our research methodology. We did not use generative AI to propose or refine hypotheses, clean or reformat data, support qualitative and thematic data analysis, or to interpret results. Creating synthetic data sets, developing theoretical models or frameworks, translation, and cleaning or reformatting data are not relevant to this work. When using AI to implement baseline methods in code, we sourced existing open-source implementations of those methods and prompted AI to adapt them for our codebase; this strategy was adopted to ensure fidelity with reference implementations. We manually verified that the implementations were correct, and when possible checked that empirical results were similar to those reported in other works. When using AI to implement our own method in code, we sketched its implementation in pseudo-code and left the LLM to fill in the blanks. We double-checked that the final implementation was in line with our original intention. The section of the paper most impacted by generative AI is the gradient analysis of §E. We prompted an LLM with a high-level description of how we believed our objective's gradient optimization was functioning; the LLM was able to confirm our intuition by deriving expressions for the gradients. While the technical details of this section are all LLM-derived, they were checked by multiple humans, and the discussion in this paper is human-authored. The entire paper is human-authored; LLMs were used to identify instances of incorrect spelling/grammar and vague or awkward wording, but any changes resulting from this

feedback were made by humans. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

B REPRODUCIBILITY

We now state details necessary to reproduce our experiments (§4). All models are trained with an ImageNet-pretrained ResNet-50 backbone. The models are trained for 100 epochs on an NVIDIA L40S GPU. For our three two-level taxonomies—living17, nonliving26, and TinyImageNet—we follow Li et al. (2024) in synthetically generating ambiguous samples by applying a Gaussian blur of kernel size three. For the remaining benchmarks, the BIOSCAN-5M subtrees, see §D. Every configuration is trained with three seeds. We will release our code upon acceptance.

Hyperparameter Grids. Each method has its own hyperparameter grid that is used across all 7 datasets. Only HRN used weight decay. When possible, we adopted the same grids as were used in a method's original published evaluation, although we diverged from these when necessary to improve performance. TREESCHISM is trained with Adam with a constant learning rate and a batch size of 64. We sweep the learning rate over $\{10^{-5}, 10^{-4}, 10^{-3}\}$ and the weight λ on the refinement gap supervision term (3) over $\{0.01, 0.1, 1\}$. HENN uses the same optimizer and batch size. We also sweep its learning rate over $\{10^{-5}, 10^{-4}, 10^{-3}\}$ and its KL regularizer's weight over $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. SoftCE and CHILLAX are trained with Adam, a one-cycle learning-rate schedule, and a batch size of 128. They sweep their learning rates over $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. HRN is trained with SGD (momentum 0.9, weight decay 5×10^{-4}), a cosine schedule, and a batch size of 128, sweeping its learning rate over $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$.

Operating points for inference procedures. As mentioned in §4, three inference procedures—THR, RAPS, and HSC—require an operating point to be specified, such as a threshold for THR and target coverage for HSC (while TREESCHISM's inference procedure uses the $\tau = 0.5$ operating point everywhere). Our experiments sweep multiple operating points on a held-out calibration set and adopt the one that maximizes the average OverJS between predicted and ground-truth labels. THR sweeps thresholds in $\{0, 0.05, 0.1, 0.15, \ldots, 0.5\}$. HSC (Goren et al., 2024) sweeps target coverage values in $\{0.05, 0.1, \ldots, 0.95\}$. RAPS (Angelopoulos et al., 2021) has three operating points: k_{reg} is swept over $\{1, 2, 5, 10, 20\}$, λ over $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 0.05, 0.2, 0.5, 0.7, 1\}$, and α over $\{0.05, 0.1, \ldots, 0.95\}$.

Model selection. For each run, we kept the model with the highest accuracy on the validation split. Across the grid, we report the configuration that produced the highest validation OverJS (defined in §4.2) averaged across three seeds. Multiple inference procedures are applied to SoftCE, CHILLAX, and HRN; we do model selection separately for each (model, inference procedure) pair. This means that, e.g., SoftCE+RAPS can select a different learning rate than SoftCE+THR.

C COMPARING TREESCHISM AND HYPER-EVIDENTIAL NEURAL NETWORKS

In this appendix, we offer a brief comparison of TREESCHISM and HENNs (Li et al., 2024) (§2). The purpose of these two methods is virtually identical: given an ambiguous sample (or "vague" sample, in the language of Li et al. (2024)), both offer a mechanism to quantify ambiguity and a predict a higher, coarser node in a taxonomy.

However, HENNs predict the parameters of a single group-Dirichlet distribution (GDD), which is a *single* distribution over all non-root nodes in a taxonomy. To model ambiguity across arbitrary taxonomies, this approach would require the use of the hyper-Dirichlet distribution. However, the full hyper-Dirichlet distribution is analytically intractable. Thus, Li et al. (2024) restrict the use cases of their method to two-level taxonomies, in which each leaf has a non-root parent, and each such parent is a child of the root. This restriction allows Li et al. (2024) to work with the tractable grouped Dirichlet distribution, a special case of the hyper-Dirichlet.

This is a density-level obstruction, not a representational one. TREESCHISM exploits this distinction. Rather than express multi-level structure inside a single density on a composite frame, we place a Dirichlet on each level in a taxonomy, and tie them together structurally through the accumulation

Table 4: **Classes per level after pruning.** A node is counted if at least one sample is labelled at or below it.

Level	Lepidoptera	Coleoptera	Hymenoptera	Hemiptera
Family	69	93	80	85
Subfamily	131	143	163	118
Genus	359	351	689	426
Species	369	398	806	413
Total	928	985	1738	1042

Table 5: **Distribution of label depth** (% of samples) in each split. The first four rows give the level of each sample's label and sum to 100. *Ambiguous* is the share of samples labelled with an internal node; it is not 100 − Species because some family, subfamily, and genus nodes are leaves. Split sizes (train / val / test): Lepidoptera 78 016 / 10 743 / 11 495; Coleoptera 186 760 / 24 406 / 25 264; Hymenoptera 386 055 / 50 427 / 52 233; Hemiptera 192 411 / 25 185 / 26 130.

	Lepidoptera			Coleoptera			Hymenoptera			Hemiptera		
Level	Train	Val	Test	Train	Val	Test	Train	Val	Test	Train	Val	Test
Family	69.0	63.4	59.9	61.4	59.2	57.7	56.8	54.6	52.9	69.4	66.7	64.7
Subfamily	3.5	3.8	4.1	28.6	27.8	27.2	19.2	18.7	18.2	6.9	6.8	6.8
Genus	14.4	15.9	16.8	4.4	5.3	5.9	16.7	17.5	18.2	13.5	14.4	15.1
Species	13.0	16.8	19.2	5.5	7.6	9.1	7.3	9.3	10.7	10.2	12.1	13.5
Ambiguous	84.7	80.1	77.3	92.6	90.1	88.3	89.0	86.6	84.8	85.7	83.4	81.7

recurrence (1). This allows TREESCHISM to generalize to arbitrarily deep taxonomies, and to localize ambiguity to any non-root node in such a taxonomy.

D BIOSCAN-5M SUBTREES

Our experiments (§4) consider four different subtrees of the full BIOSCAN-5M taxonomy. For each, we picked an order to serve as the root, and retained the family, sub-family, genus, and species levels below it. We source images from the dataset's *train* and *pretrain* splits, the latter carrying ambiguous samples without leaf labels. We discard samples annotated at the order, since the order has become the root and hence carries no information. Each node reserves two samples for each of its train, validation, and test splits; the remaining samples send 80% to train and 10% to validation and test. Any class with fewer than six samples has those samples removed; a node remains in the taxonomy only if it or one of its descendants has at least six samples. Tables 4 and 5 describe the resulting taxonomies.

These subtrees differ meaningfully in the nature of their subjects, the shape of their taxonomies, and the distribution of ambiguous samples across taxonomic levels. Lepidoptera are moths and butterflies; Coleoptera are beetles; Hymenoptera are wasps, bees, and ants; Hemiptera are true bugs. These four orders are visually distinct. Hymenoptera has far more classes than the other orders, with the surplus concentrated at the species and genus levels. After pruning, the other three orders have similar numbers of genera and species. Many species are rare and hence seldom observed; between 61% and 78% of species have their samples removed. However, this only removes 0.8%–3.3% of all samples in each order (Table 6). Family-level labels are the most common in every order, but the level that absorbs the remaining ambiguity differs across datasets. Coleoptera concentrates ambiguity at the subfamily level, with only 10% of its samples admitting a finer label; Lepidoptera and Hemiptera have fewer subfamily-level labels and more at genus and species. Hymenoptera spreads its ambiguous samples more evenly across the three non-species levels.

E GRADIENT ANALYSIS OF THE TRAINING LOSS

This appendix derives the partial derivative of the training loss (3) with respect to a single node's *local* evidence $e_m^{(\mathrm{local})}$, and examines it in three different regimes: when m lies *above* the supervision

Table 6: **Samples removed by pruning** (% of samples entering the split), i.e. samples of classes with fewer than six samples.

	Lepidoptera	Coleoptera	Hymenoptera	Hemiptera
Removed by pruning (%)	3.3	1.1	0.9	0.8

boundary (a strict ancestor of t), *at* the boundary ($m = t$), and *below* it (m a descendant of t). The analysis makes precise why the per-level classification term (i) alone cannot localize the boundary, and how the refinement-gap term (ii) supplies exactly the missing signal.

E.1 Setup

Fix a training sample with supervision boundary t at level ℓ. Write $\mathrm{sub}(n)$ for the subtree rooted at n (including n itself), $\mathrm{anc}(t)$ for the strict ancestors of t below the root, and $a_l(t)$ for t's ancestor at level $l \le \ell$ (so $a_\ell(t) = t$). Every quantity in the loss is built from the accumulated evidence (1), which unrolls to a plain subtree sum,

$$e_n^{(\mathrm{acc})} \;=\; \sum_{m \in \mathrm{sub}(n)} e_m^{(\mathrm{local})}, \qquad \alpha_n \;=\; e_n^{(\mathrm{acc})} + W_n. \tag{4}$$

Differentiating (4) gives

$$\frac{\partial e_n^{(\mathrm{acc})}}{\partial e_m^{(\mathrm{local})}} \;=\; \frac{\partial \alpha_n}{\partial e_m^{(\mathrm{local})}} \;=\; \mathbf{1}[\, m \in \mathrm{sub}(n)\,], \tag{5}$$

where $\mathbf{1}[\cdot]$ is the indicator function (1 iff its argument is true, 0 otherwise). That is, a node's local evidence enters the Dirichlet parameter of itself and each of its ancestors, and of no other node.

We use two consequences of this throughout. First, we have for the level l strength

$$S^{(l)} = \sum_{k \text{ at level } l} \alpha_k^{(l)}$$

that

$$\frac{\partial S^{(l)}}{\partial e_m^{(\mathrm{local})}} \;=\; \mathbf{1}[\,\mathrm{level}(m) \ge l\,]. \tag{6}$$

Second, $\partial \alpha_{a_l(t)}^{(l)} / \partial e_m^{(\mathrm{local})} = \mathbf{1}[\, m \in \mathrm{sub}(a_l(t))\,]$ by (5). Throughout, $\mathcal{L}_{(\mathrm{i})}$ and $\mathcal{L}_{(\mathrm{ii})}$ denote the two bracketed terms of (3) *without* the weight λ, so that $\mathcal{L} = \mathcal{L}_{(\mathrm{i})} + \lambda\, \mathcal{L}_{(\mathrm{ii})}$.

E.2 Term (i): per-level classification

Differentiating term (i) of (3) and substituting (6) gives, for any node m,

$$\begin{aligned} \frac{\partial \mathcal{L}_{(\mathrm{i})}}{\partial e_m^{(\mathrm{local})}} &= \sum_{l=1}^{\ell} \left[\frac{\frac{\partial}{\partial e_m^{(\mathrm{local})}} \left(S^{(l)}\right)}{S^{(l)}} - \frac{\frac{\partial}{\partial e_m^{(\mathrm{local})}} \left(\alpha_{a_l(t)}^{(l)}\right)}{\alpha_{a_l(t)}^{(l)}} \right] \\ &= \sum_{l=1}^{\ell} \left[\frac{\mathbf{1}[\,\mathrm{level}(m) \ge l\,]}{S^{(l)}} \;-\; \frac{\mathbf{1}[\, m \in \mathrm{sub}(a_l(t))\,]}{\alpha_{a_l(t)}^{(l)}} \right]. \end{aligned} \tag{7}$$

On-path nodes above the boundary. Suppose m is a strict ancestor of t, i.e., $m \in \mathrm{anc}(t)$ and $m \ne t$. Then $\mathrm{level}(m) < \ell$ (where, recall, ℓ is t's level). Thus, for $l \le \mathrm{level}(m)$, both indicators in (7) are 1. Hence

$$\frac{\partial \mathcal{L}_{(\mathrm{i})}}{\partial e_m^{(\mathrm{local})}} \;=\; \sum_{l=1}^{\mathrm{level}(m)} \left[\frac{1}{S^{(l)}} - \frac{1}{\alpha_{a_l(t)}^{(l)}} \right] \;\le\; 0, \tag{8}$$

since $\alpha_{a_l(t)}^{(l)} \le S^{(l)}$. Without regularization, the loss promotes local evidence on a supervised ancestor, even though we want such nodes to rely solely on accumulated evidence.

On-path nodes at or below the boundary. Here, we have $\mathrm{level}(m) \geq \ell \geq l$ for every $l \leq \ell$, so the first indicator in (7) is 1. Since $m \in \mathrm{sub}(t) \subseteq \mathrm{sub}(a_l(t))$, the second indicator is also 1 throughout. In this case, we once again have

$$\frac{\partial \mathcal{L}_{(\mathrm{i})}}{\partial e_m^{(\mathrm{local})}} \;=\; \sum_{l=1}^{\ell} \left[\frac{1}{S^{(l)}} - \frac{1}{\alpha^{(l)}_{a_l(t)}} \right] \;\leq\; 0. \tag{9}$$

The central obstruction. Term (i) is the negative logarithm of the supervised accumulated probability, and accumulated probability is blind to where in $\mathrm{sub}(t)$ its mass originates. A unit of local evidence placed on t and a unit placed on any leaf beneath t raise the supervised $\alpha^{(l)}_{a_l(t)}$ and the supervised strengths $S^{(l)}$ by exactly the same amount at every supervised level. Without additional supervision, term (i) is indifferent between t and nodes below it, and does not localize the supervision boundary to t.

Off-path nodes. Suppose $m \notin \mathrm{sub}(t)$ and $m \notin \mathrm{anc}(t)$, and let $j = \mathrm{level}(\mathrm{lca}(m,t)) < \ell$ be the deepest level at which m and t share an ancestor, so that $a_l(t)$ is an ancestor of m (equivalently $m \in \mathrm{sub}(a_l(t))$) exactly for $l \leq j$. The first indicator in (7) fires for $l \leq \min(\ell, \mathrm{level}(m))$ and the second for $l \leq j$, so

$$\frac{\partial \mathcal{L}_{(\mathrm{i})}}{\partial e_m^{(\mathrm{local})}} \;=\; \underbrace{\sum_{l=1}^{j} \left[\frac{1}{S^{(l)}} - \frac{1}{\alpha^{(l)}_{a_l(t)}} \right]}_{\leq 0,\ \text{shared supervised levels}} + \underbrace{\sum_{l=j+1}^{\min(\ell,\, \mathrm{level}(m))} \frac{1}{S^{(l)}}}_{> 0,\ \text{below the divergence}}. \tag{10}$$

Below the level at which m branches off the supervised chain, only the strength term survives and term (i) purely suppresses m's evidence. At and above that level, m shares a supervised ancestor with t, and its local evidence inflates that ancestor's accumulated mass exactly as on-path evidence would; there term (i) rewards it. A node in a different level-1 subtree ($j = 0$) never activates the second indicator: $\partial \mathcal{L}_{(\mathrm{i})} / \partial e_m^{(\mathrm{local})} = \sum_{l=1}^{\min(\ell,\, \mathrm{level}(m))} 1/S^{(l)} > 0$, yielding unconditional suppression.

E.3 Term (ii): refinement-gap supervision

Term (ii) teaches a model to identify the supervision boundary. It scores the refinement gaps along the chain of supervised nodes and enters the loss weighted by λ,

$$\mathcal{L}_{(\mathrm{ii})} \;=\; \underbrace{\mathbf{1}[t \text{ internal}]\,\big(-\log r_t\big)}_{\text{boundary term}} + \sum_{p \in \mathrm{anc}(t)} \underbrace{\big(-\log(1 - r_p)\big)}_{\text{ancestor term at } p}, \qquad r_n = \frac{e_n^{(\mathrm{local})}}{\alpha_n}. \tag{11}$$

The boundary term rewards a large gap at t (it is minimized as $r_t \to 1$), while each ancestor term rewards a small gap at p (it is minimized as $r_p \to 0$). To understand the influence of a node's local evidence on this term, we differentiate the gap $r_n = e_n^{(\mathrm{local})}/\alpha_n$ with respect to $e_m^{(\mathrm{local})}$. The local evidences $\{e_{m'}^{(\mathrm{local})}\}_{m'}$ are the independent variables of the analysis (the per-node softplus outputs), so the numerator depends on m only through n itself,

$$\frac{\partial e_n^{(\mathrm{local})}}{\partial e_m^{(\mathrm{local})}} \;=\; \mathbf{1}[\, m = n \,]; \tag{12}$$

the denominator, on the other hand, sums all of n's subtree, so by (5) $\partial \alpha_n / \partial e_m^{(\mathrm{local})} = \mathbf{1}[\, m \in \mathrm{sub}(n) \,]$. The quotient rule then gives a single expression,

$$\frac{\partial r_n}{\partial e_m^{(\mathrm{local})}} \;=\; \frac{\dfrac{\partial e_n^{(\mathrm{local})}}{\partial e_m^{(\mathrm{local})}}\,\alpha_n \;-\; e_n^{(\mathrm{local})}\,\dfrac{\partial \alpha_n}{\partial e_m^{(\mathrm{local})}}}{\alpha_n^2} \;=\; \frac{\mathbf{1}[\, m = n \,]\,\alpha_n \;-\; e_n^{(\mathrm{local})}\,\mathbf{1}[\, m \in \mathrm{sub}(n) \,]}{\alpha_n^2}, \tag{13}$$

whose value depends on where m sits relative to n (recall $n \in \mathrm{sub}(n)$):

- $m = n$. Both indicators are 1, so the numerator is $\alpha_n - e_n^{(\text{local})}$ and

$$\frac{\partial r_n}{\partial e_m^{(\text{local})}} = \frac{\alpha_n - e_n^{(\text{local})}}{\alpha_n^2} = \frac{1}{\alpha_n}\left(1 - \frac{e_n^{(\text{local})}}{\alpha_n}\right) = \frac{1 - r_n}{\alpha_n}.$$

- $m \in \text{sub}(n) \setminus \{n\}$. Only the second indicator is non-zero, so the numerator is $-e_n^{(\text{local})}$ and

$$\frac{\partial r_n}{\partial e_m^{(\text{local})}} = -\frac{e_n^{(\text{local})}}{\alpha_n^2} = -\frac{r_n}{\alpha_n}.$$

- $m \notin \text{sub}(n)$. Both indicators vanish, so the derivative is 0.

In summary, we have

$$\frac{\partial r_n}{\partial e_m^{(\text{local})}} = \begin{cases} \dfrac{1 - r_n}{\alpha_n} > 0, & m = n, \\ -\dfrac{r_n}{\alpha_n} \leq 0, & m \in \text{sub}(n) \setminus \{n\}, \\ 0, & m \notin \text{sub}(n). \end{cases} \tag{14}$$

We use this result to study the effects of term (ii) on optimization above, at, and below the supervision boundary. Throughout, we write

$$\Sigma_{\text{anc}} := \sum_{p \in \text{anc}(t)} \frac{r_p}{(1 - r_p)\,\alpha_p} \geq 0$$

for the ancestor-coupling sum.

(a) Above the boundary ($m \in \text{anc}(t)$). Differentiating $\mathcal{L}_{(\text{ii})}$ (11) term by term yields

$$\frac{\partial \mathcal{L}_{(\text{ii})}}{\partial e_m^{(\text{local})}} = -\mathbf{1}[t \text{ internal}]\,\frac{1}{r_t}\,\frac{\partial r_t}{\partial e_m^{(\text{local})}} + \sum_{p \in \text{anc}(t)} \frac{1}{1 - r_p}\,\frac{\partial r_p}{\partial e_m^{(\text{local})}}. \tag{15}$$

For a strict ancestor m, the gap derivatives (14) determine which terms survive: $m \notin \text{sub}(t)$ kills the boundary term, and an ancestor term at p survives only when $m \in \text{sub}(p)$, i.e. when $p = m$ or $p \in \text{anc}(m)$. Substituting $\partial r_m / \partial e_m^{(\text{local})} = (1 - r_m)/\alpha_m$ for the first and $\partial r_p / \partial e_m^{(\text{local})} = -r_p/\alpha_p$ for the rest,

$$\begin{aligned} \frac{\partial \mathcal{L}_{(\text{ii})}}{\partial e_m^{(\text{local})}} &= \frac{1}{1 - r_m} \cdot \frac{1 - r_m}{\alpha_m} - \sum_{p \in \text{anc}(m)} \frac{1}{1 - r_p} \cdot \frac{r_p}{\alpha_p} \\ &= \underbrace{\frac{1}{\alpha_m}}_{\text{drives } r_m \to 0} - \sum_{p \in \text{anc}(m)} \frac{r_p}{(1 - r_p)\,\alpha_p}. \end{aligned} \tag{16}$$

We have two terms. The first, $1/\alpha_m > 0$, drives m's own local evidence down as intended. The second warrants a closer look. Each summand $-r_p/[(1 - r_p)\,\alpha_p]$ is precisely $\partial[-\log(1 - r_p)]/\partial e_m^{(\text{local})}$, and the penalty $-\log(1 - r_p)$ strictly increases in r_p. Thus, the loss does not reward a larger gap. There is a genuine tension in which m's own penalty wants $e_m^{(\text{local})} = 0$, while its ancestors want to drive $e_m^{(\text{local})}$ up to inflate their own α_p without increasing the ancestor's own local evidence. However, since α_p sums all of $\text{sub}(p)$, a unit of evidence at the boundary t inflates every ancestor's α_p by exactly as much as a unit at m would. Evidence at t is rewarded by other loss terms, while evidence at m is penalized. Thus, the best way to satisfy every ancestor is to concentrate the inflating mass at the boundary and leave each strict ancestor's local evidence at zero. Indeed, setting $e_p^{(\text{local})} = 0$ for all $p \in \text{anc}(t)$ with $e_t^{(\text{local})}$ large drives every $r_p \to 0$ and $r_t \to 1$ at once, so the two objectives are jointly satisfiable with no conflict.

(b) At the boundary ($m = t$, t internal). The boundary node's own evidence is the numerator of r_t, so it drives the boundary term $-\log r_t$. The boundary node's evidence also sits in the denominator of each of its ancestors' r_p terms. Thus, increasing local evidence at t increases r_t and decreases r_p for every $p \in \mathrm{anc}(t)$. Differentiating,

$$\frac{\partial \mathcal{L}_{(\mathrm{ii})}}{\partial e_t^{(\mathrm{local})}} = \underbrace{-\frac{1-r_t}{r_t\,\alpha_t}}_{\text{drive } r_t \to 1} - \Sigma_{\mathrm{anc}} \le 0. \tag{17}$$

Both contributions are negative, confirming that term (ii) pulls evidence onto the boundary. If t is a leaf, the indicator removes the boundary term and only $-\Sigma_{\mathrm{anc}}$ survives.

(c) Below the boundary ($m \in \mathrm{sub}(t) \setminus \{t\}$). A strict descendant m is never the numerator of any gap, but since it lies in $\mathrm{sub}(t)$, it enters the denominator of r_t and of r_p for any ancestor p of t. Differentiating,

$$\frac{\partial \mathcal{L}_{(\mathrm{ii})}}{\partial e_m^{(\mathrm{local})}} = \underbrace{\frac{1}{\alpha_t}}_{\text{protect } r_t \to 1} - \Sigma_{\mathrm{anc}}. \tag{18}$$

Increasing local evidence $e_m^{(\mathrm{local})}$ below t will inflate α_t and hence lower r_t, eroding the very gap the boundary term exists to enlarge. The $1/\alpha_t$ term above penalizes descendant evidence. However, the ancestor sum $-\Sigma_{\mathrm{anc}}$ pulls in the opposite direction, rewarding evidence below the boundary.

This represents a genuine hazard for localization. Term (i) is indifferent between t and its descendants, and the ancestor coupling Σ_{anc} actively rewards pushing evidence below the boundary. The lone term resisting this effect is the boundary's $1/\alpha_t$ term. Whether the model concentrates evidence at t or leaks it below the boundary is not decided by any single term, but by the net effect of all terms. We measure the net effect in the next sub-section.

E.4 The boundary-versus-refinement wedge

Both t and its descendants $m \in \mathrm{sub}(t) \setminus \{t\}$ receive the same term-(i) gradient (9), which we denote $g_{(\mathrm{i})}$; their term-(ii) gradients are (17) and (18). Assembling the full gradient $\partial\mathcal{L}/\partial e^{(\mathrm{local})} = \partial\mathcal{L}_{(\mathrm{i})}/\partial e^{(\mathrm{local})} + \lambda\,\partial\mathcal{L}_{(\mathrm{ii})}/\partial e^{(\mathrm{local})}$ at each and subtracting, the shared $g_{(\mathrm{i})}$ and the shared coupling Σ_{anc} cancel:

$$\begin{aligned} &\frac{\partial \mathcal{L}}{\partial e_m^{(\mathrm{local})}}\bigg|_{m\in\mathrm{sub}(t)\setminus\{t\}} - \frac{\partial \mathcal{L}}{\partial e_t^{(\mathrm{local})}} \\ &= \underbrace{\Big[g_{(\mathrm{i})} + \lambda\big(\tfrac{1}{\alpha_t} - \Sigma_{\mathrm{anc}}\big)\Big]}_{\text{below } t,\ (18)} - \underbrace{\Big[g_{(\mathrm{i})} - \lambda\big(\tfrac{1-r_t}{r_t\,\alpha_t} + \Sigma_{\mathrm{anc}}\big)\Big]}_{\text{at } t,\ (17)} \\ &= \lambda\left[\frac{1}{\alpha_t} + \frac{1-r_t}{r_t\,\alpha_t}\right] = \frac{\lambda}{r_t\,\alpha_t} = \frac{\lambda}{e_t^{(\mathrm{local})}} > 0. \end{aligned} \tag{19}$$

Term (i) assigns t and each of its descendants the same gradient, while term (ii) makes the descendants' gradients exactly $\lambda/e_t^{(\mathrm{local})}$ more positive than those at the boundary. Gradient descent will therefore deposit evidence at t rather than refine past it, and the strength of that preference scales with λ and shrinks as the boundary accumulates local evidence. This is the precise mechanism through which the refinement-gap term localizes ambiguity at the supervision boundary.

Remark: absolute suppression versus relative preference. The wedge (19) is a *relative* guarantee: t is always favored over each descendant by $\lambda/e_t^{(\mathrm{local})}$. It does *not* assert that the below-boundary gradient is itself positive. Writing that gradient out,

$$\frac{\partial \mathcal{L}}{\partial e_m^{(\mathrm{local})}}\bigg|_{m\in\mathrm{sub}(t)\setminus\{t\}} = g_{(\mathrm{i})} + \lambda\Big(\tfrac{1}{\alpha_t} - \Sigma_{\mathrm{anc}}\Big), \qquad g_{(\mathrm{i})} = -\sum_{l=1}^{\ell} \frac{B_l}{S^{(l)}\,\alpha_{a_l(t)}^{(l)}} < 0, \tag{20}$$

where $B_l := S^{(l)} - \alpha_{a_l(t)}^{(l)} > 0$ is the strictly-positive off-chain α-mass at level l. Of these three pieces, two are non-positive (one strictly), so when the data-fit term is far from satisfied (i.e., $|g_{(\mathrm{i})}|$ is

large), the descendant evidence may be rewarded. This reward is, however, transient. Localize the ancestors first ($\Sigma_{\text{anc}} \to 0$, driven by their own r_p terms), so the below-boundary gradient reduces to $g_{(\text{i})} + \lambda/\alpha_t$. Now bound $|g_{(\text{i})}|$. Every strength along the supervised chain is at least α_t: each ancestor $a_l(t)$ has t in its subtree, so $e^{(\text{acc})}_{a_l(t)} \geq e^{(\text{acc})}_t$, and the leaf-cardinality prior is likewise monotone up the tree ($W_{a_l(t)} \geq W_t$); hence $\alpha^{(l)}_{a_l(t)} \geq \alpha_t$, and $S^{(l)} \geq \alpha^{(l)}_{a_l(t)}$ as a sum over its level. Every denominator in (20) is therefore at least α_t^2, so

$$|g_{(\text{i})}| \;=\; \sum_{l=1}^{\ell} \frac{B_l}{S^{(l)}\,\alpha^{(l)}_{a_l(t)}} \;\leq\; \frac{1}{\alpha_t^2}\sum_{l=1}^{\ell} B_l \;=\; \frac{B}{\alpha_t^2}, \qquad B := \sum_{l=1}^{\ell} B_l, \tag{21}$$

where B is the total off-chain α-mass along the supervised chain. The data-fit reward thus shrinks as $1/\alpha_t^2$, one power of α_t faster than the refinement penalty λ/α_t. Using $g_{(\text{i})} \geq -B/\alpha_t^2$ from (21), the net gradient obeys

$$\left.\frac{\partial \mathcal{L}}{\partial e^{(\text{local})}_m}\right|_{m\in \text{sub}(t)\setminus\{t\}} \;=\; g_{(\text{i})} + \frac{\lambda}{\alpha_t} \;\geq\; \frac{\lambda}{\alpha_t} - \frac{B}{\alpha_t^2} \;=\; \frac{1}{\alpha_t}\Big(\lambda - \frac{B}{\alpha_t}\Big) \;>\; 0 \quad\Longleftrightarrow\quad \alpha_t \;>\; \frac{B}{\lambda}. \tag{22}$$

So once the boundary's strength α_t exceeds the off-chain mass B rescaled by $1/\lambda$, the below-boundary gradient is positive and descendant evidence is actively suppressed.

Finally, note that ambiguity localization does *not* require descendant evidence to vanish entirely. We commit to t when $r_t \to 1$, i.e., when $e^{(\text{local})}_t$ dominates α_t (which the boundary term drives directly).