GREBE: A near-training-free, class-extensible scorer for passive acoustic monitoring under vocabulary and domain shift

Jack N. Etheredge[1,*]

[1]Twosense, New York, New York, United States

Abstract

BirdCLEF+ 2026, organized within LifeCLEF 2026, frames biodiversity monitoring in the Brazilian Pantanal as a closed-vocabulary, multi-label soundscape task: predict, for every five-second segment, the presence of each of 234 classes spanning birds, amphibians, mammals, reptiles, and insects. The setting is hard in two structural ways. Scoring is a domain shift: supervision is overwhelmingly *focal* (single-species clips), while the hidden test is continuous, multi-species *soundscapes*, and most classes carry no labeled soundscape audio at all. And a subset of classes, the insect *sonotypes*, lie outside the vocabulary of any pretrained classifier and can be scored from acoustic geometry alone. We set out to create a solution that is cheap to reproduce, maintain, and extend, which rules out the heavy supervised ensembles that top the leaderboard. We present **GREBE** (*Global Relative-geometry Embedding-Bank Estimator*), a nearly training-free alternative. Using Perch 2.0 strictly as a *frozen* embedder (no fine-tuning, no learned projection), we represent each class by a small bank of prototype exemplars and summarize a query window's geometry against every class bank with a compact stack of *class-relative dissimilarity features*, read by a *single global logistic regression* with no per-class parameters. Because the scoring rule is class-agnostic, one model covers both generalization regimes (the in-vocabulary focal-only species and the out-of-vocabulary insect sonotypes), and the class set is stored as a data artifact, so species are added or removed by editing exemplars rather than retraining. One round of self-training on the unlabeled soundscape pool is the single largest performance step. The resulting single model reaches 0.917 public / 0.923 private macro ROC–AUC (an optional per-class environmental extension reaches 0.931 private), within roughly 0.04 private of the winning entry among 4,091 teams, while training only one logistic regression over cached embeddings on CPU in seconds. We report our original four-model ensemble as an upper bound, and show that it proved unnecessary: a single model distilled from the ensemble's pseudo-labels lands within 0.003 public / 0.005 private of it, and a variant that instead generates its pseudo-labels from one of two lightweight scorers built on the frozen embedder, chosen per species, reaches the same macro ROC–AUC without the ensemble at all. For our pipeline the ensemble was needed neither at inference nor to generate the pseudo-labels. Code is available at https://github.com/Jack-Etheredge/birdclef2026.

Keywords

BirdCLEF+, Perch, Embedding, Multi-label, Pseudo-label, Efficient, Passive acoustic monitoring, Soundscapes, Biodiversity

1. Introduction

Passive acoustic monitoring (PAM) has become a central tool for tracking biodiversity across regions that are too large, too remote, or too dynamic to survey by hand. Continuously recording the soundscape of a habitat captures the vocalizations of many co-occurring species at once and yields a community-level view of how an ecosystem changes through seasons and disturbance. BirdCLEF+ 2026 [1] draws its recordings from the Brazilian Pantanal, one of the world's largest wetlands, home to over 650 bird species alongside amphibians, mammals, reptiles, and insects. The Pantanal is increasingly stressed by seasonal flooding, fire, agricultural expansion, and climate change. This is exactly the kind of landscape where passive monitoring matters: a growing network of acoustic recorders now runs continuously across the region, but the volume of audio it produces vastly exceeds what experts can annotate. The

CLEF 2026 Working Notes, Friedrich-Schiller-Universität Jena, Germany, from September 21–24, 2026

*Corresponding author.

jack.etheredge@gmail.com (J. N. Etheredge)

https://github.com/Jack-Etheredge (J. N. Etheredge)

0000-0001-5467-3866 (J. N. Etheredge)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

bottleneck is no longer recording; it is turning unlabeled audio into reliable, species-level evidence at scale. The challenge, organized within the LifeCLEF 2026 lab [2], poses this as a concrete machine-learning problem: given one-minute soundscapes recorded by the Pantanal PAM network, predict, for each non-overlapping five-second segment, the presence of each of 234 classes spanning five taxonomic groups (Aves, Amphibia, Mammalia, Insecta, and Reptilia).

Two properties make the task hard in ways that matter for real deployment. First, the evaluation is a *domain shift*: almost all of the supervision is *focal* (short, largely single-species clips contributed to Xeno-Canto and iNaturalist), while scoring is on continuous, multi-species *soundscapes* drawn from the monitoring network. Crucially, only a minority of classes have any labeled soundscape audio at all; the majority carry focal recordings only, so a usable scorer must transfer across the focal-to-soundscape gap with no in-domain supervision for most of the taxonomy. Second, a subset of the classes (the insect *sonotypes*, which are not resolved to species) lie outside the output vocabulary of any available pretrained classifier, and can be scored only from acoustic geometry or from the few in-domain examples that exist. These two difficulties are disjoint problems over disjoint sets of species: one is *out-of-domain* transfer within a closed label space (focal labels, no soundscapes), and the other is *out-of-vocabulary* scoring (no pretrained head at all). Compounding both is a practical constraint shared by the recent BirdCLEF series [3, 4]: a tight, CPU-only inference budget that penalizes heavy pipelines. Beyond the metric, a real monitoring program also demands a solution that is cheap to reproduce, maintain, and extend as new species are encountered in the field.

The strongest leaderboard entries answer the performance question with weight: deep supervised mel-spectrogram detectors, large ensembles, inference-time model fusion, and iterative self-training on the unlabeled soundscapes [5, 6, 7, 4]. These pipelines are accurate, but their cost is paid not only in compute but in *reproducibility, maintainability, and extensibility*: a fixed classification head bakes the class count into the trained parameters, so adapting to a new site or a newly discovered species means retraining. Prior BirdCLEF+ 2025 solutions bracket this design space at a heavy end and a lightweight end (Section 2); the gap they leave open is the one we target: a method as cheap as the lightweight lineage but competitive in macro ROC–AUC with the heavy one.

Our approach occupies that middle. We use a broadly pretrained bioacoustic embedder, Perch 2.0 [8], strictly as a *frozen* feature extractor (no fine-tuning, no learned projection), and rely on its documented ability to transfer to taxa and domains nearly absent from pretraining [9, 10, 11]. Each class is represented by a small bank of prototype exemplars in the frozen embedding space, in the spirit of distance-based and prototypical classifiers that absorb new classes at near-zero cost [12, 13]. Rather than scoring a query window by a single nearest-prototype similarity, we summarize its geometric relationship to each class bank with a compact stack of *class-relative dissimilarity features* and read them with a *single global logistic regression*: one weight vector shared across all 234 classes, with no per-class parameters and no backbone training. Because every feature is defined relative to the class and its competitors rather than in absolute embedding coordinates, the same scoring rule applies uniformly to classes the model was never fit on, so one scorer covers both the out-of-domain (no-soundscape) species and the out-of-vocabulary insect sonotypes without any per-class machinery or routing rule. The result is competitive with far heavier pipelines while being dramatically cheaper to train, reproduce, and deploy, and (because the class set is stored as a data artifact scored by a class-agnostic rule) trivially extensible: adding or removing a species means adding or dropping exemplars, not retraining.

This lean design did not come first: our original entry was a heavy four-model ensemble (the frozen-embedder scorer plus a supervised mel-spectrogram sound-event detector, a sequence model over the embedding stream, and an environmental prior, refined by self-training on the unlabeled soundscapes), and the present work is a deliberate effort to find how much of that apparatus could be removed without giving up its score. We therefore report the ensemble only as an *upper bound*, not a contribution, and Section 5.5 shows it is unnecessary both at inference and as the source of the pseudo-labels.

The primary contributions of this paper are:

1. **A nearly training-free scorer for soundscape classification.** We turn a frozen audio embedder into a stack of class-relative dissimilarity features read by a single global logistic regression (no

fine-tuning, no learned projection, and no per-class parameters), and isolate this design against the off-the-shelf nearest-neighbour and masked-logit recipes that the embedder's authors recommend.
2. **One model for two distinct generalization regimes.** The same scorer handles in-vocabulary species with no labeled soundscapes (focal-to-soundscape transfer) and out-of-vocabulary insect sonotypes (scored on geometry alone), covering the full closed taxonomy with a single class-agnostic rule rather than a per-class classifier or hand-built routing.
3. **A cost–performance characterization of the heavier alternatives.** We quantify how much leaderboard performance the additional ensemble models and self-training yield over the lightweight scorer, and which of those gains are portable to a real field deployment rather than artifacts of the competition metric.
4. **An honest offline-to-leaderboard evaluation protocol.** Because the standard offline surfaces systematically anti-track the hidden test in this transfer setting, we treat the leaderboard as the final arbiter for ranking design choices and use the offline surfaces (including a no-soundscape surface for the untransferred species, judged on site-grouped folds) as calibration and screening signals where informative rather than as the primary metric.

2. Related work

We organize the related work around the components our method builds on and the prior solutions it positions itself against.

2.1. The spectrogram–CNN paradigm and its cost

The dominant approach to bioacoustic classification, and to BirdCLEF in particular, converts each recording to a mel-spectrogram and applies image-style convolutional backbones, frequently with a weak-label attention head for frame-level prediction, with the strongest entries ensembling several such detectors [5, 6, 7, 4]. BirdNET [5] and the large-scale audio-tagging networks pretrained on AudioSet (PANNs) [6] are representative of this lineage. While accurate, deep supervised ensembles are expensive to train, to run within the competition's inference deadline, and (more importantly for real-world deployment) to reproduce, maintain, and extend, since a fixed classification head bakes the class count into the trained parameters. Our work deliberately steps away from this paradigm in favor of a frozen embedder and a lightweight scorer.

2.2. Frozen bioacoustic embedders and cross-domain transfer

A complementary line of work shows that large bioacoustic models trained on broad species inventories yield embeddings that transfer well to new tasks, taxa, and acoustic environments. The Google Bird Vocalization Classifier / Perch and BirdNET produce general-purpose birdsong embeddings that support strong transfer learning [9, 5], and the BirdSet release adds Torch-native backbones benchmarked under a common protocol [11]. Most relevant to us, Perch 2.0 [8] is a multi-taxa successor whose frozen embeddings have been shown to transfer even to domains almost absent from pretraining (Burns et al. report strong few-shot linear-probe transfer to marine-mammal tasks, frequently beating specialized marine models [10]), and cross-domain mixing during pretraining was earlier shown to be the key to such transfer for coral-reef audio [14]. Our approach uses Perch 2.0 strictly as a frozen feature extractor, with no fine-tuning and no learned projection, and relies on exactly this transfer property to score species that the embedder was never trained to name.

2.3. Prototype, metric-learning, and few-shot scorers

Distance-based classifiers score a query by its proximity to per-class exemplars in a fixed embedding space, and generalize to new classes at near-zero cost because adding a class only adds exemplars rather than retraining a head [12]. Prototypical networks formalize this for few-shot learning [13],

and learnable-scale cosine classifiers extend it to settings where new classes must be absorbed without forgetting [15]; Perch 2.0 itself ships an intrinsic prototype-style head [8]. Our scorer sits in this family but departs from the usual single-distance rule: it describes a query by its *dissimilarities* to each class's prototypes and classifies in that relational space, placing it in the classical dissimilarity-representation lineage [16, 17] and its bioacoustic instantiation, where sounds are classified by distances to per-class centroids [18]. Two things set it apart from that line: the dissimilarities are read off a *frozen, off-the-shelf* embedder with no representation learning (that line trains Siamese networks), and a *single global* readout is shared across all classes rather than per-class classifiers, which is what lets one class-agnostic rule span the out-of-vocabulary and out-of-domain regimes. Classifying each query by its similarity to a stored exemplar set makes the approach instance-based and metric-based in character. It remains *inductive*: the exemplar banks and the shared readout are fit once and then applied to unseen soundscapes, and the system is scored on a held-out test set. That is distinct from the *transductive* objective recently raised for ecological labeling, where the aim is to annotate an entire collected pool as efficiently as possible rather than to predict on held-out data [19]. It is also related to centroid- and cache-based adapters in vision [20, 21], which, however, score by raw similarity rather than by a learned readout over relative geometry.

2.4. Domain shift, pseudo-labeling, and semi-supervised learning

Because BirdCLEF scores on out-of-domain soundscapes, mitigating the focal-to-soundscape shift is central to strong solutions, and since unlabeled test-domain soundscapes became available the standard lever has been semi-supervised self-training on them. Pseudo-labeling [22] and its confidence-adaptive refinements [23, 24] underpin recent entries: the BirdCLEF+ 2025 runner-up uses iterative pseudo-label distillation as its single largest source of gain [7], and earlier DS@GT work frames soundscape pseudo-labeling as a transfer-learning surrogate task [25]. Our approach also exploits the unlabeled soundscapes, but routes pseudo-labeled windows only into the global linear scorer for labeled classes, leaving the frozen embedder and the prototype banks untouched, a far cheaper form of self-training than retraining a supervised backbone or distilling into a separate student.

2.5. Class-extensibility versus open-set recognition

Keeping the class set in a data artifact scored by a class-agnostic rule lets species be added or removed with no retraining, the property the agile-modeling line for bioacoustics is built on, where field-discovered species are folded in within an hour via embedding search and a lightweight classifier [26, 10]. We emphasize that this is class-extensibility, not open-set recognition [27, 28]: the task remains a closed 234-class problem with no novelty rejection, and the class set is simply a deployment-time input rather than an architectural dimension. This distinction matters for our positioning, since the practical value we claim is cheap extensibility within a fixed label space, not detection of unknown categories.

2.6. Prior-year archetypes

The two strongest BirdCLEF+ 2025 working notes bracket the design space our method targets. At the heavy end, Sydorskyi and Gonçalves placed second (0.928 private) with a five-part supervised pipeline (strong CNN baselines, in-domain backbone pretraining on hundreds of thousands of recordings, iterative pseudo-label distillation, temporal postprocessing, and tuned ensembling) and documented that local cross-validation decorrelates from the leaderboard above roughly 0.9 public, alongside several prominent negative results [7]. At the lightweight end, the DS@GT team studied off-the-shelf model-zoo transfer under the CPU budget and proposed STSG, a spectrogram-token skip-gram model that runs in minutes but scores only 0.559 public, while noting that macro ROC-AUC saturates and must be supplemented by a more discriminative metric [29]. These are prior-year systems (BirdCLEF+ 2025 covered 206 classes in the Middle Magdalena Valley of Colombia [4], whereas 2026 covers 234 classes in the Pantanal [1]), so we treat them as method archetypes and sources of transferable methodological

Table 1
The canonical class partition: vocabulary (in-vocabulary IV vs out-of-vocabulary OOV) against domain (in-domain ID, with labeled soundscapes, vs out-of-domain OOD, focal-only). The OOV+OOD cell is empty, so the two generalization regimes are disjoint species sets. These counts are canonical; all other bucket cuts in the paper derive from them.

	ID (labeled soundscapes)	**OOD** (focal-only)
IV (in vocabulary)	50	159
OOV (no logit)	25	0 (empty)

findings rather than as directly comparable score baselines. Our approach aims for the unfilled middle: as cheap as the lightweight lineage but competitive in macro ROC–AUC with the heavy one, evaluated under the same offline-to-leaderboard discipline that both prior solutions found necessary.

3. Dataset and transfer structure

3.1. Task and provenance

BirdCLEF+ 2026 [1, 2] draws its data from the Brazilian Pantanal (Mato Grosso do Sul), a wetland of more than 150,000 km^2 surveyed by a growing passive-acoustic-monitoring (PAM) network. Within its fixed 234-class label space, bird classes are keyed by eBird species codes and non-bird classes by iNaturalist taxon IDs; most insect classes are not resolved to species and instead enter as *sonotypes* (e.g. `47158son16`) that are treated as their own classes. All audio is distributed as 32 kHz OGG. Supervision comes in two forms with very different character: *focal* recordings (short, largely single-species clips contributed to Xeno-Canto and iNaturalist) and *soundscape* recordings captured by the same PAM network that produced the hidden test set. At inference, roughly 600 one-minute test soundscapes are scored in non-overlapping five-second segments, and a presence probability is required for each of the 234 classes in every segment; the private leaderboard is computed on approximately 66% of the test data. Recorder sites span −16.5° to −21.6° latitude and −55.9° to −57.6° longitude.

3.2. Two axes, one empty cell: the canonical partition

Two properties of a class jointly decide which scorers apply to it and where transfer is required. The first is **vocabulary** status: a class is *in-vocabulary* (IV) if the frozen embedder exposes an output logit for it and *out-of-vocabulary* (OOV) if it does not. The second is **domain** status: a class is *in-domain* (ID) if it has labeled soundscape audio and *out-of-domain* (OOD) if it has only focal recordings. Cross-tabulating these two axes splits the 234 classes into a 2×2 with one empty cell (Table 1); throughout, we call each of the four quadrants of this 2×2 a *bucket*.

Because the OOV+OOD cell is empty, the two transfer problems never coincide in a single class. Out-of-domain transfer is needed for the **159** in-vocabulary species with no labeled soundscapes (the *no-soundscape* species), while out-of-vocabulary handling is needed for the **25** insect sonotypes, which carry soundscape labels but no embedder logit. The soundscape-labeled set therefore numbers **75** (50 + 25), and the rest of the taxonomy carries no in-domain labels at all. By embedder vocabulary, 203 of the 234 classes carry a direct Perch logit, 6 fall back to a same-genus proxy, and the 25 sonotypes have none.

3.3. Taxonomic and genus structure

The partition is highly uneven across taxa (Appendix A, Tables 10–12). The no-soundscape species are overwhelmingly avian (**134 of 159**, 89%), and only 15 Aves, 3 Mammalia, and 1 Reptilia clear the $\geq$ 5-window floor needed to fit a class directly, so per-taxon fitting is data-starved for three of the five taxa and must pool or shrink across regimes. The species are also genus-sparse (most genera

Table 2
Audio volume by source (five-second windows). Soundscapes are 60 s (12 windows each); focal recordings average 7.0 windows. The 739 labeled windows are the only in-domain supervision.

Source	Recordings	5 s windows	Role
Focal (`train_audio`)	35,549	**233,101**	out-of-domain reference; 206 in-set species only
Unlabeled soundscapes	10,592	**127,104**	in-domain self-training pool (pseudo-labels)
Labeled soundscapes	66	**739**	the *only* in-domain supervision

contribute one species), and genus provides almost no bridge between the labeled in-domain set and the no-soundscape species: only **13%** of out-of-domain species (21 of 159) have a labeled congener (6% within the avian bulk). The lone exception is Amphibia at 72%, so genus is unusable as a *global* transfer axis but a genuine *local* lever for amphibians, exactly where environmentally structured (diel- and rain-driven) calling makes an environment-aware prior most informative.

3.4. Supervision asymmetry

The asymmetry that motivates the whole approach is the scarcity of in-domain labels (Table 2). Focal audio is abundant (233,101 five-second windows across 35,549 recordings) but is out-of-domain, and it covers only the 206 in-set species that have focal material (the 28 soundscape-only species have none). The unlabeled soundscape pool adds 127,104 in-domain windows but no labels; it is the substrate for self-training. Against these, the *only* in-domain supervision is 739 labeled windows from 66 soundscapes, roughly 315× scarcer than the focal material and 170× scarcer than the unlabeled pool, and leaving the no-soundscape species with zero in-domain labels. The class distribution is also strongly long-tailed in recordings per class. A frozen embedder, prototype geometry, and pseudo-labeling (Section 4) are the natural response to this regime.

3.5. Label density and multi-label structure

The 739 labeled windows are multi-label, averaging **4.22** species/window (std 1.78; the 5th, 25th, 50th, 75th, and 95th percentiles are 1, 3, 4, 5, and 7; range 1–10). This mean sets the self-training admission target (Section 4.6).

3.6. Site and time coverage

Recordings carry a site tag and a UTC timestamp, and the labeled and unlabeled soundscapes differ sharply in coverage (Appendix A, Table 13). Two asymmetries drive the validation design. First, a **site mismatch**: the labeled set is dominated by a single site (S22, 65% of labeled windows, 477 of 739), while the pool's two largest non-S22 sites (S01 and S02, together ≈46% of pool windows) carry essentially no labels, and 14 of the pool's 21 sites never appear in the labeled set. The labeled surface is therefore not representative of the deployment distribution, which is why we judge offline on site-grouped folds and treat the public leaderboard as the arbiter (Section 4.9). Second, a **narrow time match**: labeled and unlabeled soundscapes agree in time (both overwhelmingly night/crepuscular, with almost no midday recording), so the solar/diel band is nearly constant across the data, consistent with rain, which varies independently of time of day, being the strongest environmental signal while the crepuscular bands add little. Taxon is moreover almost perfectly confounded with site in the labeled data: each labeled site captured essentially one taxonomic community (S22, 65% of the labels, is 98% Amphibia), so any taxon×environment pattern read off the pooled labels is really a site×environment artifact (Table 14). Two consequences shape the environmental head (Section 4.6): its pool-learned profiles are estimated from the unlabeled pool rather than the site-confounded labels, and of its two hand-set interactions

only Aves×diel is corroborated by the labeled data, while the Amphibia×rain prior rests on external ecology. Appendix A gives the full density-structure and contingency detail.

4. Methodology

Building on the dataset and transfer structure of Section 3, we describe our approach in the order the pipeline runs: the frozen embedder and its logits (Section 4.1), the per-species prototype banks (Section 4.2), the class-relative dissimilarity features (Section 4.3), the single global scorer and how it is fit (Section 4.4), and the two generalization regimes one scorer covers (Section 4.5). We then summarize the full ensemble system the scorer anchors (Sections 4.6–4.8) and the validation and evaluation protocol (Section 4.9). We call this core scorer **GREBE**, a dissimilarity prototype scoring model (expanded in the abstract), and use that name for it throughout. Figure 1 sketches the pipeline end to end.

Every component is implemented in PyTorch [30] or scikit-learn [31]: the supervised sequence and detection models (Section 4.6) are PyTorch, while the global scorer and the per-class environmental head (Section 4.7) are scikit-learn estimators. The frozen Perch 2.0 and BirdNET embedders are run as ONNX exports with ONNX Runtime [32] for CPU-inference speed relative to their PyTorch versions. All training reported in this paper was run on a single desktop workstation with one NVIDIA RTX 5090 GPU.

4.1. Frozen embedder and its logits

We use Perch 2.0 [8] strictly as a *frozen* feature extractor, producing a 1536-dimensional embedding per five-second window. The backbone is never fine-tuned, and we deliberately apply *no* learned projection, whitening, or metric-learning head on top of the embeddings: a learned projection was tested and did not improve results, so the raw embedding space is used directly. Mean pooling over the window is sufficient; adding max pooling or concatenating BirdNET [5] embeddings did not improve ranking performance. Choosing a frozen, broadly-pretrained embedder is what makes the method portable across taxa and sites: Perch and related bioacoustic embedders are known to transfer well even to taxa and domains nearly absent from pretraining [9, 10, 11].

In addition to the embedding, Perch 2.0 exposes a per-class classification logit for every species in its output vocabulary. Where a class c has such a logit, we read its *provenance-masked* value (the activation of c's own output unit only) and use it as a single additional feature (Section 4.3). For the 6 *near-vocabulary* classes (no direct logit, but a same-genus species in the vocabulary), we substitute the max over the same-genus logits as a weak proxy; the 25 out-of-vocabulary sonotypes contribute no logit and are scored on geometry alone. Using the embedder's own logit where it exists, and falling back to geometry where it does not, is what lets one scorer span the full taxonomy.

4.2. Prototype banks

Each class is represented by a small bank of *prototype* exemplars in the frozen embedding space. For class c with n_c positive windows we run k-means++ over its available embeddings (focal-clip windows plus any labeled soundscape windows, all L_2-normalized) to obtain a set of centroids $\{p_c^{(1)}, \dots, p_c^{(K_c)}\}$. The number of centroids is set to half the class's window count and clamped to a guardrail range, $K_c = \max\big(1, \min(10, \lfloor n_c/2 \rfloor)\big)$ (i.e. $\lfloor n_c/2 \rfloor$ clusters, floored at one and capped at ten), so a class with three or fewer windows collapses to a single prototype while one with twenty or more is held at ten; k-means++ then partitions all of the class's windows into those K_c clusters. The cap of ten is a fixed heuristic default chosen as a sensible value rather than tuned: we did not sweep it, and a cap above ten was not tested (the CH-adaptive variant of Section 5 searches only up to the same ceiling), so we make no claim that ten is optimal; smaller per-class counts simply arise automatically for sparse classes. Rare species get no special scoring rule. The per-window similarities are used as *features* by the single global classifier (Section 4.4) shared across all classes, and a rare species differs only in contributing fewer prototypes (a smaller K_c). Each centroid is the Euclidean mean of its assigned

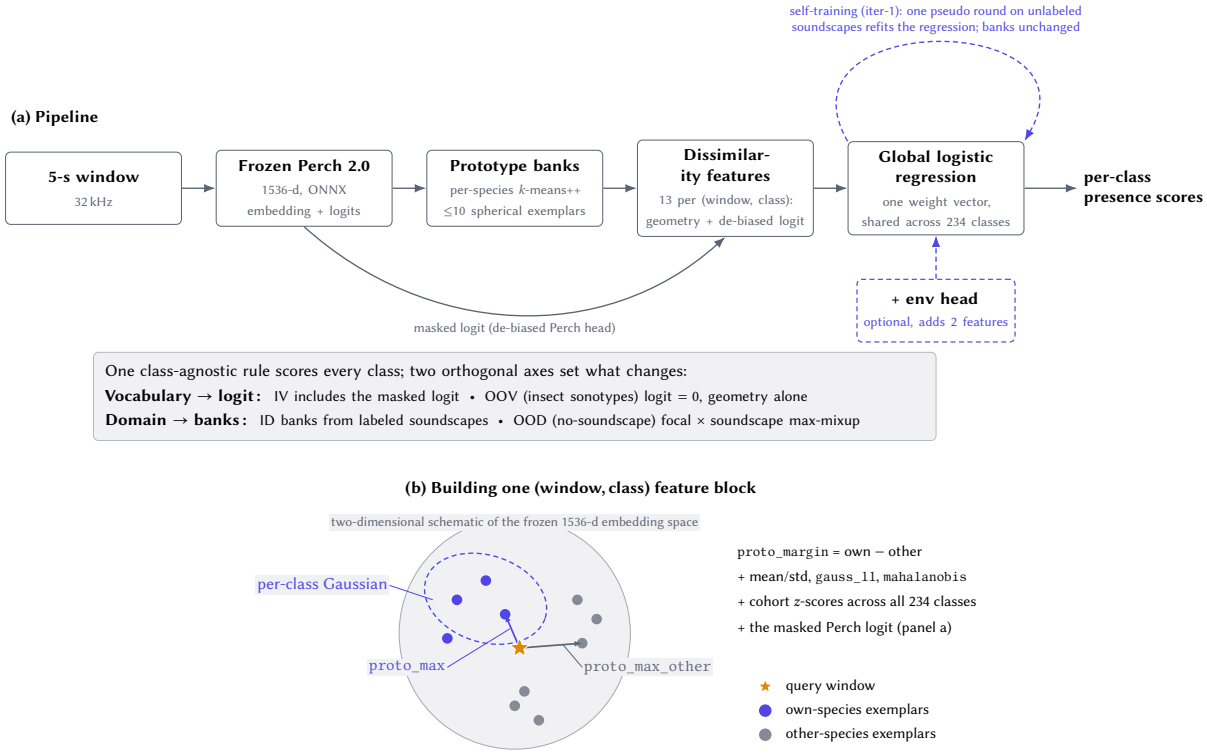

Figure 1: The GREBE pipeline. **(a)** A 5-s window is embedded once by a frozen Perch 2.0 model; per-species prototype banks (*k*-means++ exemplars, at most ten per class, scored against spherical-mean centroids) and the model's own masked, de-biased logit yield 13 class-relative dissimilarity features per (window, class). A single global logistic regression, one weight vector shared across all 234 classes, maps those features to per-class presence scores. One class-agnostic rule scores every class; two orthogonal axes set what changes per class. Vocabulary, whether a class is in-vocabulary (IV) or out-of-vocabulary (OOV), governs whether the Perch logit is available; domain, whether it is in-domain (ID) or out-of-domain (OOD), governs how the banks are built (labeled soundscapes versus a focal-by-soundscape max-mixup for the no-soundscape species). Self-training adds one pseudo round that refits only the regression, leaving the banks unchanged (Section 4.6); the environmental head is an optional extension that appends two features (Section 4.7). **(b)** The Dissimilarity-features step for a single query window. Each feature is a *relative* geometric quantity, not a thresholded decision: the cosine similarity to the nearest own-species exemplar (`proto_max`), the similarity to the nearest other-species exemplar (`proto_max_other`), their margin, the mean and spread over the class's own exemplars, a per-class Gaussian (`gauss_ll`, `mahalanobis`), and cohort *z*-scores that rank each score across all 234 classes. The same shared classifier reads this feature block for every class, so a rare species is handled by the identical rule with fewer exemplars, never by a species-specific similarity cutoff.

(unit-norm) embeddings, re-normalized to the unit sphere (a spherical-mean prototype, not a geometric mean), and a query is scored against the bank by maximum cosine similarity. Banks are built with a fixed random seed; the k-means step runs on GPU when one is available and on CPU otherwise, which yield equal-inertia but not bit-identical centers, a backend difference we hold constant and gate on the offline labeled-soundscape OOF surface (Section 4.9). Representing a class by a handful of exemplars rather than a trained head is what makes the class set a *data artifact*: adding or removing a class means adding or dropping exemplars, with no retraining [12, 13].

To bridge the focal-to-soundscape gap for the no-soundscape species (the IV+OOD bucket, whose species have only focal audio and no soundscape labels of their own), their focal exemplars are combined with soundscape *background* embeddings by an element-wise **max-mixup**,

$$\tilde{p} = \max\left(p^{\text{focal}},\ p^{\text{SC-bg}}\right) \quad \text{(element-wise)},$$

which empirically outperforms linear embedding-space or waveform mixup as a bank construction for the learned scorer (Section 5). We additionally fit a per-class diagonal Gaussian over the exemplars

to supply likelihood- and Mahalanobis-based features (Section 4.3). All embeddings are L_2-normalized within the view before similarities are computed, so every similarity below is a true cosine.

4.3. Class-relative dissimilarity features

Rather than score a query window x by a single nearest-prototype similarity, we summarize its geometric relationship to each class bank with a fixed stack of **thirteen** *class-relative* features, read jointly by the global scorer (Section 4.4). Let $e(x)$ be the unit-normalized embedding of x. For class c, recall its bank of K_c k-means++ exemplars $\{p_c^{(1)}, \dots, p_c^{(K_c)}\}$ (Section 4.2) and its per-class diagonal Gaussian $\mathcal{N}(\mu_c, \operatorname{diag}\sigma_c^2)$. Write $s_c = \{\cos(e(x), p_c^{(k)})\}_{k=1}^{K_c}$ for the K_c exemplar cosine similarities, and let the *cohort* be the set of classes scored for this window. Every feature below is computed from s_c and the Gaussian; nothing uses absolute embedding coordinates. The thirteen features for the pair (x, c), in five **per-class**, five **cross-class**, two **cohort-normalized**, and one **logit** term, are:

- *per-class geometry* (class c's own fit): the maximum $\text{proto_max}_c = \max_k s_c[k]$ (the cosine 1-nearest-neighbour), the mean $\text{proto_mean}_c = \operatorname{mean}_k s_c[k]$, and the standard deviation $\text{proto_std}_c = \operatorname{std}_k s_c[k]$ of the exemplar similarities; the diagonal-Gaussian log-likelihood $\text{gauss_ll}_c = -\frac{1}{2}\big(\sum_d (e(x)_d - \mu_{cd})^2/\sigma_{cd}^2 + \sum_d \log \sigma_{cd}^2\big)$; and the squared diagonal Mahalanobis distance $\text{gauss_maha}_c = \sum_d (e(x)_d - \mu_{cd})^2/\sigma_{cd}^2$;
- *cross-class competition* (how c stands against the field): the best competitor's prototype-max $\text{proto_max_other}_c = \max_{c' \neq c} \text{proto_max}_{c'}$ and Gaussian log-likelihood $\text{gauss_ll_max_other}_c = \max_{c' \neq c} \text{gauss_ll}_{c'}$; the prototype margin $\text{proto_margin}_c = \text{proto_max}_c - \text{proto_max_other}_c$ and Gaussian margin $\text{gauss_margin}_c = \text{gauss_ll}_c - \text{gauss_ll_max_other}_c$; and the gap to the runner-up, $\text{proto_max_minus_top2}_c = \text{proto_max}_c - (\text{2nd-largest proto_max over all classes})$;
- *cohort normalization* (class c's standing within the whole window's class cohort, a Z/T-norm): $z\text{-pmax}_c = (\text{proto_max}_c - \mu_x)/\sigma_x$ and $z\text{-gll}_c = (\text{gauss_ll}_c - \mu'_x)/\sigma'_x$, where (μ_x, σ_x) and (μ'_x, σ'_x) are the mean and standard deviation of the corresponding feature taken across the cohort for that window; and
- the *logit* term: the provenance-masked Perch logit for c (Section 4.1), *de-biased per class* by subtracting a fixed per-class floor q_c (the 25th-percentile logit of class c computed once over the unlabeled soundscape pool, a leak-free, test-domain estimate), then standardized over all logit-present (window, class) entries, and set to 0 for the out-of-vocabulary sonotypes that have no logit. The subtraction is applied identically at fit and inference; it recenters each class's logit onto its own noise floor and, empirically, is worth +0.0033 private on the leaderboard (Section 5.1).

Because the cross-class and cohort terms define every feature *relative to the class and its competitors* rather than in absolute embedding coordinates, the same feature definitions apply to classes the scorer was never trained on: a window's score for c depends on c's *standing among the competitors*, not on its raw geometry. That is precisely what lets one global rule transfer to the no-soundscape and out-of-vocabulary regimes. The cross-class standardization is closely related to score-normalization techniques in speaker verification [33, 34].

4.4. Global logistic regression scorer

All classes are scored by a *single global* logistic regression (LR) over the feature stack of Section 4.3; there are no per-class coefficients and no per-class shrinkage. The same weight vector w and bias b are applied uniformly to all 234 classes, including the no-soundscape species. Sharing one coefficient vector is a deliberate strong-regularization choice: relaxing it to per-class scorers was tested and did not pay off in general (the per-class few-shot probe loses to the shared readout, Section 5.2.1), the one exception being the per-class environmental head, which helps only for the environmental signal (Section 5.4.2). Nowhere does the system classify by thresholding a cosine similarity: the similarities enter only as features to this regression. Two normalization steps make the geometry comparable across features

and views. First, the per-view L_2 normalization of Section 4.2 ensures each similarity is a scale-free cosine. Second, the regression is fit on standardized features (zero mean, unit variance per column), and the scaler is *folded into the deployed coefficients*,

$$w = \frac{\text{coef}}{\sigma}, \qquad b = b_0 - \sum_j \frac{\text{coef}_j \, \mu_j}{\sigma_j},$$

so that no feature (and in the multi-view case no view) can dominate by raw scale, while deployment remains a single linear layer. Inference scores are globally rank-averaged, and every evaluation metric is computed per class (Section 4.9).

Out-of-fold fitting (stacked generalization). The prototype banks and the global regression are fit on *disjoint* routings of the same labeled data so the regression sees realistic held-out geometry. For each of five folds k, the bank is built from the training folds ($\neq k$) and the held-out fold k is scored against that bank to produce leak-free out-of-fold (OOF) features; pooling the OOF features over all folds, we fit one global regression. At deployment the regression is used as-is, but with the prototype banks rebuilt once on *all* available data (no fold ensembling). Thus the deployed banks are fold-independent while the deployed regression coefficients depend on the OOF feature distribution, which is why the choice of fold scheme (Section 4.9) affects the leaderboard even though the banks are identical.

Fitting and class balance. The scorer is a scikit-learn `LogisticRegression` (`lbfgs` solver, $C = 1.0$, `max_iter` $= 500$) fit over the standardized features. Its training rows are pooled (held-out window $\times$ labeled class) pairs: every labeled class is emitted for every OOF window, and the binary target is whether that window carries that class. A window typically carries only about four of the roughly sixty scored labeled classes, so the absent-class pairs dominate. We handle this imbalance with class weighting rather than resampling: we neither under-sample the absent pairs nor over-sample the present ones, and instead set scikit-learn's `class_weight` to `balanced`, which up-weights each class inversely to its frequency. Inference is then a single standardized linear layer: for each (window, class) pair the thirteen features are computed, the folded scaler and coefficients of the equation above are applied, and the resulting per-class scores are globally rank-averaged before evaluation.

Pseudo-labels refit the scorer, not the banks. When self-training is used (Section 4.6), pseudo-labeled pool windows enter *only* the global-LR fit as additional training rows, carried at a reduced per-row `sample_weight` (ground-truth rows at 1.0, a pseudo row at pseudo_weight $\cdot \hat{y}$) multiplied on top of the balanced class weights. They never become prototype exemplars: the k-means banks of Section 4.2 are always built from ground-truth (focal and labeled-soundscape) audio only. We explored adding pseudo-labeled windows as exemplar keys and it consistently regressed performance, so the banks stay ground-truth-only and pseudo-labels act purely as scorer supervision.

4.5. Two generalization regimes, one scorer

The construction above handles both generalization problems identified in Section 3.2 with one model. For **out-of-domain** classes (focal labels but no labeled soundscapes), the class-relative features are computed against focal-derived, domain-bridged banks and read by the same global regression that was fit on soundscape-labeled classes; no in-domain labels are needed for the no-soundscape species. For **out-of-vocabulary** classes (the insect sonotypes, with no Perch logit), the masked-logit feature is simply absent and the class is scored on geometry alone. No separate model, routing rule, or per-class fitting is introduced for either regime; the genus-proxy logit (Section 4.1) is the only class-specific adjustment, and it targets just six near-vocabulary classes.

4.6. The four-model starting point

Our single global scorer began as the anchor of a heavier *four-model ensemble*, the configuration that produced our highest leaderboard entries and the starting point for this work. That ensemble is a hybrid: the dissimilarity scorer, the environmental components, and the learned blend are our own, while the supervised SSM and SED models are near-identical reproductions of public community notebooks (detailed below). We summarize it here and treat it throughout as a heavy *upper bound* whose cost the rest of the paper sets out to remove, not as the contribution; the main thrust is the simpler, cheaper, more deployable single model that replaces it (Section 5.1). The models are:

- **the dissimilarity scorer** of Sections 4.1–4.5 (anchor; covers all taxa);
- an **SSM model**, where SSM denotes a selective state-space model: a prototypical sequence model over the per-window Perch embeddings combining a Mamba-style selective state-space backbone [35, 36], a Transformer cross-attention block [37] for non-local temporal structure, per-class prototypes [13, 15], and a gated fusion of the Perch logit. This model is a near-identical reproduction of a public community Kaggle notebook [38]; we include it primarily because it *decorrelates* from the anchor (correlation $\rho \approx 0.72$);
- a **SED model**, where SED denotes sound-event detection: a supervised mel-spectrogram detector with its own 234-class output heads, the strongest *individual* model on the no-soundscape surface, and likewise a near-identical reproduction of a public community Kaggle notebook [39];
- an **environmental scorer**: a single per-class histogram-gradient-boosting (HGBC) model of $P(\text{present} \mid \text{environment})$ that sees only raw environmental signals (no acoustics): solar elevation and a rising/setting flag (derived from the recorder's coordinates and UTC timestamp via a standard solar-position approximation [40]), plus temperature, humidity, cloud cover, and rain at several lags (from the Open-Meteo historical API [41]). A class with at least thirty pool positives gets its own model; the rest fall back to a neutral score. This one scorer is consumed two ways. (i) In the four-model ensemble its per-class score grid is blended directly in rank space as an independent model (carried mainly on Insecta). (ii) In the *unified* single-model lineage we put forward, the very same per-class scorer is used not as a separate model but as a *feature*: its output, together with an external diel/rain anchor, is fed into the global dissimilarity regression (the per-class presence head of Section 4.7). The two are thus one scorer with two consumption paths, differing only in whether its grid is rank-blended against other models or read as an input to the single global linear model. Distinct from this scorer is the *environmental admission gate*: it draws on the *same raw signals* but *not* the HGBC model, applying a multiplicative environmental factor by separate logic when reweighting candidates in the pseudo-labeling loop. The gate was used only for the four-model teacher ensemble, and under the logit+dissimilarity teacher that produces our headline model it gave no improvement (results not shown).

The models are combined in rank space by a *learned* per-(taxon × bucket) weighting rather than hard routing, so a model receiving zero weight on a cell (e.g. the supervised models on no-soundscape cells, which have no in-domain labels to learn from) is a learned outcome, not an architectural rule. The rank-average blend itself is the community recipe popularized by the SSM notebook above, which combined two models (SSM and SED) at a fixed 60 : 40 weight tuned on the public leaderboard; our blend instead learns the weights of all four models from out-of-fold scores, on the same file-respecting folds used throughout (Section 4.9), rather than fixing a two-model split on the leaderboard. An optional pseudo-labeling self-training loop [22, 7] uses a teacher to admit high-confidence windows from the unlabeled soundscape pool, refits the trainable models, re-weights, and iterates. We index rounds as *iter-n*: **iter-0** is the no-self-training starting point (no pool windows used), **iter-1** the model after one round of pseudo-labeling, and **iter-2** after a second round. The admission policy depends on the teacher. The four-model ensemble teacher admits a (window, class) pair when the class clears a per-class noise-floor margin (its score exceeds the class's own low-score quantile by $\tau = 0.2$), clears an absolute confidence floor of 0.3, and is among the window's top $k = 5$ classes, with the environmental and

SED models further gating the decision. Centering each class on its own noise floor makes detections comparable across classes despite per-class baseline drift; it is the one-sided analog of the Z/T-norm score normalization noted in Section 4.3. The single-model lineage that we put forward (Section 5.1) instead uses a lighter per-bucket teacher: the masked logit for the no-soundscape species and the dissimilarity scorer for the in-domain buckets. For each teacher a single *global probability* threshold (one cutoff applied to the teacher's per-class predicted probabilities, the same value for every class) is calibrated so the admitted pool averages 4.22 species per window, matching the labeled prior (Section 3.5); a flat per-species cap of at most 128 admitted windows per class is then applied to the windows that clear the threshold. For honest offline judging the teacher is *fold-isolated* (each pool window scored by a teacher blind to its fold) to avoid leaking held-out labels, whereas the *deployed* labels use an all-data teacher. A structural property separates the models for real-world use: the dissimilarity and env models are **class-extensible** (their class set is stored as a data artifact scored by a class-agnostic rule), whereas the SSM and SED models bake the class count into trained parameters and must be retrained to change the label set.

4.7. Environmental extension: a per-class presence head

The deployed environmental signal in the single-model lineage is a per-class **presence head**: a histogram-based gradient-boosting classifier (HGBC), the scikit-learn `HistGradientBoostingClassifier` [42]. It is an *extension* on top of the headline scorer, not part of it: the headline single-agile model (Section 5) carries no environmental feature, and the head is reported as a ceiling on what environment can add. The exhaustive comparison of alternative environmental encodings (per-species marginals, per-class exemplar geometry, and shared-coefficient typicality features, all of which proved neutral or harmful) is deferred to Section 5; here we describe the one form that helps.

What the head consumes. For each class c we fit one binary HGBC that predicts $P(\text{class } c \text{ present} \mid \text{env})$ from the nine z-scored environmental signals introduced in Section 4.6: solar elevation, a solar rising/setting indicator, temperature, precipitation at the window (`rain_now`), precipitation at -3 h, -6 h, and $+3$ h lags, relative humidity, and cloud cover. These are computed per window from the recorder's coordinates and UTC timestamp (solar elevation via a standard solar-position approximation [40]; the weather variables and rain lags from the Open-Meteo historical API [41]). The HGBC is chosen deliberately over a linear head: the presence-versus-environment relationship is non-linear (for example a nocturnal species is a near step-function of solar elevation), and a per-class logistic head over the same inputs was net-negative on both evaluation surfaces (Section 5).

How it enters the scorer. The head's scalar output, env_p_present, is appended to the global-LR feature stack of Section 4.3 as one extra feature, alongside a second feature a_{ext}, an external diel/rain anchor. a_{ext} is hand-set from each species' published diel descriptor (nocturnal, diurnal, or crepuscular, each obligate or facultative) crossed with the window's solar band, plus a single taxon override (Amphibia in rain, a rain-chorus prior); it is leak-free, since it is not estimated from the soundscapes, and it reaches all 234 classes including those with no logit. The two features extend the deployed model from thirteen to fifteen inputs of the same single global logistic regression; there is still one shared coefficient vector, with env_p_present supplying the per-class environmental information.

How the environmental pool is built. The heads are fit on the union of the labeled soundscape windows and the pseudo-labeled windows admitted by the self-training round (Section 4.6); a window counts as a positive for class c when c is one of its ground-truth or pseudo labels. Each class's training pool is capped at 128 windows (the "flat-128" admission). The cap matters: fitting on the uncapped pool was noisier and weaker, because the coverage is floor-limited rather than density-limited (removing the cap adds windows only to classes that are already data-rich, none to the sparse tail). A class needs enough positive windows to fit its own head; 184 of the 234 classes clear that support floor, and the

remainder fall back to a single pooled head fit across all classes, which still returns a real environment-informed probability rather than a constant. As with the prototype banks and the global regression, the head is fit two-track to avoid leakage: a fold-isolated out-of-fold column supplies the env_p_present values that the global regression is fit on, while a full-pool head is shipped in the deployment bundle. At inference the nine signals are recomputed per window, env_p_present is read from the deployed head (the class's own where it cleared the support floor, else the pooled fallback) and a_{ext} from the species descriptor, and the same global regression scores the fifteen-feature vector, adding no separate model beyond one cached per-class probability and a table lookup.

4.8. Postprocessing

Deployed scores optionally pass through a short postprocessing chain: temporal smoothing across a soundscape's windows (a vocalization-continuity prior) and a file-level rescaling that scales each window by a function of its file's top-K window scores (which sharpens the cross-file ranking the macro-AUC metric rewards). We report results both with and without the chain, and separate its deployment-valid and competition-only contributions in Section 5.3.2.

4.9. Validation, splits, and evaluation

Splits. We use a five-fold cross-validation in which whole files are held out (no recording's windows are split across folds in either scheme), comparing a **site-grouped** scheme (folds grouped by recording site) against a **species-stratified** (file-grouped, multilabel-stratified) scheme. Because the labeled soundscapes are dominated by one site (Section 3.6), file-grouped folds leak that site's acoustics across folds and read optimistically, while site-grouped folds are the honest *judge*. Empirically, however, the species-stratified OOF features are a *better fit target* for the deployed global regression. We therefore **judge offline on site-grouped folds but fit/deploy the scorer on species-stratified folds**, and treat the public leaderboard as the final arbiter for ranking design choices.

Evaluation surfaces. We use three surfaces and report whichever are informative for a given result, rather than all three on every row. (i) A *labeled-soundscape OOF* surface: out-of-fold macro AUC over the real ground-truth-labeled soundscapes, which covers only the in-domain (ID) species. (ii) A *no-soundscape* surface for the out-of-domain species, which have no labeled soundscapes of its own: it estimates per-class AUC on the no-soundscape Aves by a file-grouped per-species cross-validation of their focal clips, with the element-wise max-mixup banks of Section 4.2 standing in as the focal-to-soundscape proxy for these out-of-domain species (file-grouped so that near-duplicate windows of one recording never appear in both bank and queries). (iii) The Kaggle leaderboard (public and private), which is the surface we report for most results; the two offline surfaces are used as calibration and screening signals where noted.

Metric and significance. The competition objective is a macro-averaged per-class ROC-AUC that omits any class with no positive labels in the scored window set [1]; every offline and leaderboard number in this paper uses this metric. Because macro ROC-AUC saturates near the top of the range in this transfer setting [29], verdicts use paired per-class bootstrap confidence intervals against pre-registered noise floors rather than raw score deltas. Alongside the score we report a cost vector (parameters trained, GPU-hours, and number of models at inference) for the configurations we compare on cost (Table 8), to make the efficiency trade-off explicit [4].

Decision rule. All gating, ranking, and tuning decisions are made on offline surfaces or the *public* leaderboard. The private leaderboard is recorded on every run for honest final reporting and to show public/private agreement, but is never used as a decision metric.

Table 3
The self-training lineage of the single agile model (macro ROC–AUC, mean ± std over five joint seeds). Both iter-2 rows are on the base recipe, before the logit de-biasing.

Stage	Configuration	LB public	LB private
iter-0 (no pseudo)	best no-pseudo baseline	0.8954 ± 0.0012	0.9084 ± 0.0006
iter-1 (one pseudo round)	single global LR, raw logit	0.9110 ± 0.0024	0.9201 ± 0.0032
+ **de-biased logit (primary)**	+ per-class logit de-biasing	**0.9167 ± 0.0018**	**0.9234 ± 0.0025**
+ env head (overall SOTA)	+ per-class env. (HGBC) head	0.9221 ± 0.0021	0.9309 ± 0.0023
iter-2 (second round)	second self-training round	0.9032 ± 0.0024	0.9149 ± 0.0021
iter-2 + env	second round, env-headed teacher	0.9033 ± 0.0032	0.9148 ± 0.0033

5. Evaluation performance and ablation studies

Our aim is not a leaderboard placement but a quantified efficiency argument: how close the near-training-free single-model scorer of Section 4 comes to far heavier pipelines, and what each expensive component actually contributes. The central result is a short *self-training lineage* of that model (Section 5.1). Everything after it (scorer baselines, mixup operators, embedder views, fold schemes, postprocessing) is supporting evidence that the simple design was chosen against tested alternatives, not by default. The four-model ensemble is treated throughout as a deliberately out-of-scope upper bound (Section 5.5).

Baseline, splits, and seeds. We follow the evaluation protocol of Section 4.9. Unless stated otherwise the baseline for an ablation is **GREBE-alone** (a single model, not the ensemble) on frozen Perch embeddings with the deployed prototype-bank settings, its global scorer fit out-of-fold under stacked generalization. GREBE has two stochastic knobs: the exemplar bank (k-means++ and mixup sampling) and the fold partition. Unless noted, we report each arm's total variability as a mean ± standard deviation over a shared joint-seed axis {42, 1337, 404, 9001, 91}.

5.1. Main result: one agile model and its self-training lineage

The deployed system is a single global logistic regression over class-relative features on a frozen Perch 2.0 embedder, refined by one round of self-training and a per-class de-biasing of the logit. Its score is reached along a short lineage that also marks its own ceiling and its point of diminishing returns (Table 3). We treat this lineage as the paper's central result, and each later subsection provides the evidence that one of its steps was justified.

iter-0: the no-pseudo baseline. Without any self-training, the best single-model routing scores the masked Perch logit on the no-soundscape species and the GREBE geometry on the soundscape-labeled species (the *logit-OOD/GREBE-ID* routing justified in Section 5.2.1), reaching 0.8954 public / 0.9084 private. This is the honest floor of the lineage: a frozen embedder, exemplar banks, and one global regression, with no pool windows touched.

iter-1 and the primary model. One round of self-training over the unlabeled pool is the single largest step. Pool windows are admitted by the per-bucket, ground-truth-label teachers described in Section 4.6 (the masked logit for the no-soundscape out-of-domain species, the dissimilarity scorer for the in-domain buckets). The *same* global regression is refit on labeled plus pseudo rows at full pseudo-weight, moving the model from 0.9084 to 0.9201 private (+0.0117). One further scorer-side change completes the deployed *primary* model: de-biasing the masked-logit feature per class (subtracting each class's low-quantile logit computed once over the unlabeled pool, Section 4.3) adds an orthogonal +0.0033 private / +0.0057 public, to 0.9167/0.9234. This is the configuration we put forward as the paper's result, and it remains one agile single model: no ensemble and no per-class parameters (Section 5.5).

Table 4
GREBE versus off-the-shelf Perch-recommended scorers, all on frozen Perch embeddings. Macro ROC–AUC, mean ± std over five joint seeds; the leaderboard is the arbiter for the no-soundscape species. GREBE clears the best *simple* baseline (raw-bank cosine-kNN) by +0.0154 public and beats logit-alone by +0.069.

Scorer	Recipe origin	LB public	LB private
GREBE (geometry + masked logit, global LR)	ours	**0.8881 ± 0.0045**	0.8941 ± 0.0046
GREBE, fold-ensembled scorer	ours	0.8883 ± 0.0045	0.8942 ± 0.0045
GREBE, genus-proxy logit off	ours	0.8874 ± 0.0045	0.8932 ± 0.0046
GREBE, geometry only (logit dropped)	ours	0.8642 ± 0.0057	0.8609 ± 0.0057
cosine-kNN, raw focal banks	Perch	0.8727 ± 0.0017	0.8779 ± 0.0017
soft-kNN (τ=0.1), raw focal banks	classic	0.8725 ± 0.0014	0.8797 ± 0.0007
cosine-kNN K=4 (Perch 4-proto), mixup banks	Perch	0.8554 ± 0.0037	0.8546 ± 0.0093
cosine-kNN CH-adaptive K, mixup banks	ours	0.8582 ± 0.0050	0.8546 ± 0.0050
cosine-kNN K=10, mixup banks	—	0.8502 ± 0.0051	0.8483 ± 0.0077
soft-kNN (τ=0.1), mixup banks	classic	0.8472 ± 0.0025	0.8444 ± 0.0064
cosine-kNN K=20, mixup banks	—	0.8460 ± 0.0071	0.8464 ± 0.0058
masked logit alone	Perch	0.8196 ± 0.0000	0.8127 ± 0.0000

+ env: a lightweight extension. Adding a per-class histogram-gradient head that predicts presence from each window's environmental metadata (solar elevation, rain lags, temperature, humidity, cloud cover), fed as two extra features to the global regression, stacks additively on the de-biased-logit primary and lifts the ceiling to 0.9221/0.9309 (+0.0054/ + 0.0075; the logit-de-biasing and environmental gains are on orthogonal features and add cleanly). This is the overall single-model SOTA; it exceeds the full four-model ensemble (0.914/0.928) on both surfaces, though that cross-split comparison (five-seed species-split versus single-seed site-split ensemble) is only suggestive, not a controlled head-to-head. We keep it separate from the primary model because the head is *per class* (trading the single-model, class-extensible property) and its gain is site-correlated rather than field-stable; Section 5.4.2 gives the details.

iter-2: saturation. Re-admitting the pool with the iter-1 model as the new teacher and refitting does *not* compound: both the env-off and env tracks regress to $\approx$ 0.915 private, and, tellingly, the 0.0063 iter-1 gap between them closes to a tie. The first pseudo round is the whole self-training win; a second saturates and regresses regardless of track. We report iter-2 precisely to mark that the lineage stops at one round, and to show the env advantage does not survive being hardened into pseudo-labels (Section 5.4.2).

5.2. Supporting evidence I: what the scorer earns over off-the-shelf Perch

The lineage rests on a scorer that must beat the obvious ways to use the embedder. We compare GREBE against the masked per-class logit alone, cosine nearest-neighbour over the prototype bank (equivalently Perch's intrinsic prototype head), and soft-kNN, on both raw-focal and domain-bridged (max-mixup) banks. All arms share the same frozen embeddings, the same banks, and the same cosine metric; only the *scoring* differs. The labeled-only 16-shot linear probe cannot cover the no-soundscape species and so produces no full submission; its inability to address the out-of-domain majority is itself part of the argument for a class-relative scorer. Results are five joint seeds (Table 4).

Read as a contribution ladder, no single ingredient explains the result. The masked logit alone reaches 0.820 (rank-valid, but at chance on the out-of-vocabulary insect sonotypes, which have no Perch logit); a cosine nearest-neighbour over the prototypes reaches 0.873; the richest single component, GREBE's own geometry stack without the logit, reaches 0.864; and GREBE (the global regression over the geometric features *and* the masked logit) reaches 0.888. The logit is worth +0.024 to GREBE (0.864 $\rightarrow$ 0.888), but geometry-only with the logit removed still beats every mixup-bank nearest-neighbour scorer. So neither reductive reading survives: GREBE is not "the Perch logit with overhead" (logit-alone trails by

Table 5
Per-bucket scorer selection. Each entry is a paired five-seed isolation of one scorer against GREBE on that bucket alone (every other class masked to chance, probe given all positives, postprocessing off); the value is the winner's private margin and the fraction of seeds it leads. "—" = not applicable (the no-soundscape species have no soundscape labels for a probe; the insects have no Perch logit). The per-bucket winners *are* the deployed logit-OOD/GREBE-ID routing (the no-soundscape species to the logit, the ID labeled species to GREBE).

Bucket	Classes	logit vs GREBE	probe vs GREBE	Deploy
IV+OOD (no soundscape)	159	logit +0.0142 (5/5)	—	logit
IV+ID (in-vocab, SC)	50	GREBE +0.0062 (5/5)	GREBE +0.0055 (5/5)	GREBE
OOV+ID (insects)	25	—	GREBE +0.0028 (5/5)	GREBE

0.069), nor "a nearest-neighbour retriever" (the best NN trails by 0.015). A single linear layer combines a strong-but-saturated zero-shot signal with a noisier-but-complementary class-relative geometry into a score that beats either in isolation.

5.2.1. GREBE beats per-class few-shot probes once the comparison is fair

The strongest off-the-shelf way to use Perch is *per species*: the masked logit where a class is in-vocabulary, an in-domain few-shot probe where it has soundscape labels. Isolating logit-versus-probe on the species where both apply, the 16-shot probe wins by +0.0031 public (Appendix B, Table 17): under domain shift a class is better served by a few in-domain examples than by the zero-shot logit. That does *not*, on its own, tell us how the probe compares to *GREBE*: a naive macro comparison is confounded because the composite and GREBE differ in *two* places at once (the composite falls back to the zero-shot logit on the no-soundscape species, where GREBE uses geometry). We therefore isolate each bucket instead, holding that routing fixed.

To set the deployed routing, we isolate each of the three buckets and pit each available scorer against GREBE directly, masking every other class to chance and giving the probe *all* positive windows per class (uncapped, unlike the composite's 16-shot probe), paired across five seeds with postprocessing off (Table 5). The picture is unambiguous and is the source of the logit-OOD/GREBE-ID routing. On the no-soundscape species, which have no labels for a probe, the zero-shot logit beats GREBE (+0.0142 private, all five seeds), so these species deploy to the logit. On the in-vocabulary soundscape-labeled bucket all three scorers are available, and GREBE wins both pairings, over the all-positives probe by +0.0055 (per-class effect +0.026, diluted in the masked macro) and over the bare logit by +0.0062, so the ranking is GREBE > probe > logit. On the insect sonotypes, which have no Perch logit, GREBE beats the probe on private (+0.0028, all five seeds) and ties on public. The earlier apparent GREBE-probe parity was thus an artifact of the capped, bucket-mixed probe; isolated and uncapped, GREBE *outscores* the per-class probe wherever both apply. This is why the deployed system carries *no* per-class probes and routes every soundscape-labeled bucket through the single GREBE regression, the logit-OOD/GREBE-ID routing that is iter-0 of the main lineage (Table 3).

5.3. Supporting evidence II: design choices we tested

The remaining in-body controls settle how the global scorer is fit and how the postprocessing chain is chosen; each follows the same pattern (a tested alternative the leaderboard rejected), and several expose an offline↔leaderboard inversion that motivates our selection discipline. Two further bank- and embedder-level ablations (the prototype mixup operator and multi-view embeddings) point the same way and are deferred to Appendix B.

5.3.1. Cross-validation split, scorer fit, and seed structure

Three closely related controls settle how the global scorer is fit. First, the *fold scheme* (Table 6). The labeled soundscapes are dominated by a single recording site, so a site-grouped holdout reads

Table 6
Fold scheme, five seeds (all postprocessing on). Species-grouped folds win the leaderboard and are fully separated from site-grouped; site-grouped is retained only as the honest offline judge.

Fold scheme	LB public	LB private
Species-stratified (deploy)	**0.8911 ± 0.0049**	**0.8930 ± 0.0048**
Site-grouped (judge)	0.8750 ± 0.0043	0.8833 ± 0.0051
Δ (species − site)	+0.0161	+0.0097

Table 7
Postprocessing screen (GREBE-alone, single seed, public). The deployed chain pairs a *deployment-valid* step (temporal smoothing, which encodes vocalization continuity and transfers to any continuous recording) with a *competition-only* one (file-level rescaling, a cross-file macro-AUC re-ranker that does not transfer); per-step contributions are in the Δ column.

Configuration	LB public	Δ vs all-off
All on	**0.8848**	+0.0118
Drop file-rescale	0.8817	+0.0088
All off	0.8730	—

pessimistically while a species-stratified (file-grouped) holdout lands on the leaderboard level. We pre-registered that site-grouped folds would *win* the board; they did not. Species-grouped folds win by +0.0161 public over five seeds, with the two seed distributions fully separated. The resolution is to use *both*: judge offline on site-grouped folds (the honest, harsher estimate) but fit and deploy on species-grouped folds, because the leak-inflated out-of-fold features are a better *fit target* for the deployed regression even though they are the worse judge.

Second, the *stacked-generalization discipline*. Fitting the deployed regression on out-of-fold features versus in-sample, all-data features is worth +0.06 on the leaderboard (0.9005 versus 0.8429 public, single seed). At inference a test window is never in the bank, so its features are held-out-like; the in-sample fit learns weights calibrated to self-matched geometry that does not occur at test. The out-of-fold fit is essential, not a formality.

Third, the *seed structure*. GREBE's two stochastic knobs behave oppositely across surfaces. Offline, the fold partition looks like the dominant lever and the bank roughly ten times smaller; on the leaderboard the ranking reverses, and the bank seed is about 2.6× more variable than the fold seed (public std 0.0051 versus 0.0020). This is another instance of the offline surface anti-tracking the board, and it is why we report total variability on a joint-seed axis and confirm axis winners on the leaderboard rather than on a single offline draw.

5.3.2. Genus-proxy logit and postprocessing

Two smaller controls round out the design. For the near-vocabulary species, the *genus-proxy logit* takes the max over their same-genus logits. Masking it shifts the leaderboard by +0.0007, within the noise floor, so it is neutral. The *postprocessing* chain (temporal smoothing and file-level rescaling) is on in every leaderboard number reported elsewhere; a three-arm screen isolates its contribution (Table 7). The chain is net-positive and non-trivial (+0.0118), of which roughly +0.0088 is field-portable (temporal smoothing encodes vocalization continuity) and roughly +0.0031 is a leaderboard-only macro-AUC re-ranker (file rescaling). Which half survives a field deployment is itself a result (Section 5.5).

5.4. Supporting evidence III: the pseudo round and its environmental head

This subsection documents the two steps that carry the lineage past iter-0: the single pseudo round (iter-1) and the optional per-class environmental head.

5.4.1. Why one pseudo round, and at what weight

A single round over the unlabeled pool, with pool windows entering the global-LR fit and banks unchanged, is the entire self-training win (iter-0→iter-1 in Table 3). The pseudo-weight is a real lever (the admitted pool is roughly three times the ground-truth soundscape cell count): sweeping it with the ground-truth rows fixed is monotone on *both* surfaces with no per-seed sign flips (Appendix B, Table 16), so full weight is the operating point. At low weight the no-soundscape bulk prefers logit routing; at full weight the all-GREBE scorer overtakes the logit there, which is why the iter-1 model fits every class with one regression rather than routing.

Offline, the mechanism is legible on the no-soundscape surface: the pseudo round leaves the labeled-soundscape surface flat (+0.000) but lifts the no-soundscape surface by +0.110 (0.704 → 0.814, single seed). The pseudo rows refine the *global* regression on labeled classes, and the class-relative features transfer the refinement to the no-soundscape species, the only ones for which real soundscape signal was otherwise unavailable.

5.4.2. The per-class environmental head (iter-1 + env)

The environmental extension adds two features to the global regression: a per-class histogram-gradient model of presence given the environmental signals, and an external diel/rain anchor that reaches all classes. Among head families and training-pool caps (five seeds), the capped per-class gradient-boosted head is the both-surface winner: on the raw-logit recipe it adds +0.0063 private. Because it is a feature orthogonal to the logit de-biasing, the two gains add cleanly: colz (+0.0033) and env (+0.0063) over iter-1 predict +0.0096, and the observed +0.0108 (to the overall single-model ceiling of 0.9221/0.9309) is fully additive, even a hair super-additive, and comes with *lower* variance. A linear head of the same form is net-negative, and training on the uncapped pool underperforms the cap (the coverage is floor-limited, not density-limited). We hold this apart from the primary model for two reasons. The signal is *discriminative and per-class*: it is captured only by a separate small model per class, so adopting it trades iter-1's single-model, class-extensible property for one head per class. And it is *site-correlated rather than field-stable*: no class shows cross-site-consistent environmental preference, so the head generalizes on the held-out leaderboard but is not guaranteed to transfer to new sites, and, as iter-2 shows, the advantage does not survive being hardened into pseudo-labels and re-fit (the env and env-off tracks tie after a second round). It is a genuine, cheap score-improving extension; it is simply a different efficiency claim, so we report it as one.

5.5. Cost↔performance and leaderboard standing

The efficiency thesis is that the primary model gives up a small leaderboard margin for large practical wins. Table 8 places each configuration's macro ROC–AUC next to its cost vector: parameters trained, training GPU-hours, and number of models at inference. The primary model trains only a single global logistic regression over cached features on top of a frozen embedder, with no per-class parameters and no backbone training; the environmental extension adds one small gradient-boosted head per class; the supervised models bake the class count into trained parameters and must be retrained to change the label set. We sharpen the claim two ways. First, *field-portability*: the postprocessing steps and the environmental head split by whether they transfer to a real deployment. Temporal smoothing transfers to new sites and hours because it encodes vocalization continuity, and it carries the bulk of the postprocessing gain (+0.0088 of the +0.0118 chain, Table 7); the competition-only file rescaling (+0.0031) and the site-correlated environmental gain are macro-AUC re-rankers that do not. Second, *class-extensibility*: the primary model keeps its class set in a data artifact scored by a class-agnostic rule, so a field-discovered species folds in by adding exemplars with no scorer retrain [8, 10], a property the per-class environmental head and the supervised models lack.

The seed-54 deploy: the ensemble's value collapses into pseudo-labels. The full 4-model entry comes from a *single* run that *predates every other experiment in this paper*: it is our original deployed

Table 8
Cost ↔ performance; score is public/private macro ROC–AUC (rounded to three decimals, with the precise five-seed means in Tables 3 and 4). Train GPU-h is one 5-fold (re)train on the project's local GPU: the single-model configs are near-training-free (a k-means bank and one logistic regression over cached embeddings), while the 4-model figure is SED-bound (~130 min run concurrently on 4–5 GPUs). The top block is the multi-seed lineage; the bottom three rows are a *single-seed* (seed 54) matched ladder (no pseudo, then one ensemble-taught pseudo round, then the three non-GREBE models added), reported to illustrate the distillation and not directly comparable to the multi-seed means above.

Configuration	Params trained	Train GPU-h	Models @ inf.	LB pub / priv
Off-the-shelf Perch logit	0	0	1	0.820 / 0.813
iter-0, no pseudo (logit-OOD/GREBE-ID)	~13 (global LR)	< 0.1	1	0.895 / 0.908
primary (single agile model, de-biased logit)	~13 (global LR)	< 0.1	1	**0.917 / 0.923**
primary + env head	~13 + 1 HGBC/class	< 0.1	1	0.922 / 0.931
GREBE-alone, no pseudo (seed 54)	~13 (global LR)	< 0.1	1	0.898 / 0.907
GREBE-alone, + ensemble pseudo (seed 54)	~13 (global LR)	< 0.1	1	0.911 / 0.923
Full 4-model ensemble (seed 54)	~13 (LR) + ProtoSSM + SED (neural)	~10.3	4	0.914 / 0.928

competition ensemble, and the starting point from which we set out to find a simpler, near-training-free model that matches it. It is a single self-training round (iter-1) on the site-grouped split at seed 54, in which the *full four-model ensemble itself serves as the pseudo-label teacher* and pool windows are admitted by a per-class noise-floor margin ($\hat{s}_c = s_c \cdot \text{env_factor} - \text{floor}_c$, admit if $\hat{s}_c$ clears the per-class threshold) with a multiplicative environmental *gate* on admission. Toggling the models at inference on this one run (the *same* teacher, labels, split, and seed) scores 0.914/0.928 with all four models and 0.911/0.923 with the dissimilarity model alone (the seed-54 matched pair, Table 8). Two conclusions follow, and together they are the core efficiency argument of the paper.

(1) Most of the ensemble survives distillation into a single global model. On this same single-seed, site-grouped basis the lone dissimilarity model scores 0.898 public with no self-training; trained on the ensemble's own pseudo-labels it rises to 0.911 (+0.013), within +0.003 public / +0.005 private of the 0.914/0.928 four-model ensemble it was distilled from (Table 8, seed-54 ladder). The pseudo round, taught by the ensemble, closes roughly four-fifths of the 0.016 public gap between the bare single model and the full ensemble. So almost all of the ensemble's macro ROC–AUC can be captured *once*, offline, as pseudo-labels and then discarded: what deploys is one logistic regression over a frozen embedder, not four neural models at inference. The ensemble's role becomes that of a one-time teacher, not a serving-time dependency.

(2) The ensemble is not needed even as the teacher. We then removed it from the teaching loop entirely: the multi-seed primary model (Table 3) uses only *ground-truth-label* teachers (the masked logit and the GREBE-SC scorer, no ensemble) on the species-stratified split, and reaches the same public score ($\approx$ 0.911) by this ensemble-free route. The seed-54 run is a different recipe from the primary model (noise-floor-margin, env-gated admission, site-grouped split, single seed), and our controlled comparisons find that recipe no better than the cleaner flat-cap ground-truth-label teacher admission, with the env gate flat against it (Sections 5.4.1, 5.4.2). We therefore present seed 54 as a single-seed illustration of the distillation, not a recommended operating point.

Leaderboard standing. Table 9 situates our configurations against the published BirdCLEF+ 2026 leaders (N=4,091 teams). We do not claim a placement; the point is the trade. The leaders reach higher scores with heavy supervised detectors, deep multi-model ensembles, inference-time augmentation,

Table 9
Leaderboard context (public / private macro ROC–AUC), top five of 4,091 teams by private score, shown against our configurations; the comparison is cost-for-performance, not a placement claim. Every "Ours" configuration runs on the same frozen Perch 2.0 embedder, so the Stack column lists only the components added on top. The lineage rows are multi-seed (n=5) means; the two seed-54 rows are a single-seed matched pair, not directly comparable to them.

Entry	Stack	LB pub / priv
Nikita Babych	—	0.9672 / 0.9657
tennogh	—	0.9600 / 0.9601
kapenon	—	0.9509 / 0.9599
BirdCLEF+ 2026 Team	—	0.9633 / 0.9590
Jiacheng Ma	—	0.9572 / 0.9582
Ours, 4-model ensemble (seed 54, matched)	SSM + SED + env	0.914 / 0.928
Ours, GREBE-alone (seed 54, matched)	global LR + pseudo	0.911 / 0.923
primary + env head (5-seed, overall SOTA)	global LR + pseudo + HGBC/class	0.922 / 0.931
primary (5-seed, de-biased logit)	global LR + pseudo	**0.917 / 0.923**
iter-0, no pseudo (5-seed)	global LR	0.895 / 0.908

and self-training loops that are correspondingly harder to reproduce, maintain, and deploy, whereas the single-model primary configuration comes within a small margin on the same metric (about 0.04 private of the winning entry, and about 0.035 with the environmental extension) at a fraction of the training and inference cost. (Public and private orderings differ noticeably among the leaders, e.g. the fourth-place private team holds the second-highest public score, underscoring why we select on public and report private rather than the reverse.)

5.6. Forward-looking directions

Several alternatives were tested and rejected, each reported where it appears (in the sections above or in Appendix B). Looking ahead, the high-priority items are the per-class environmental head as a learned *deploy* model on the teacher's own admissions rather than a post-hoc feature, per-cell weighting granularity against a global blend, and a broader comparison of readouts over the exemplar similarities (our single global linear head is one point in that space).

6. Conclusions

We set out to find the minimal solution that is viable for a real passive acoustic monitoring deployment, and the answer is a single, nearly training-free model: a frozen Perch 2.0 embedder [8], a small bank of k-means++ prototype exemplars per class, and one global logistic regression over class-relative geometric features. Because the scoring rule is global and class-agnostic, the same model handles every combination of in-domain, out-of-domain, in-vocabulary, and out-of-vocabulary species; the only requirement on a class is that it carry at least one of an in-vocabulary logit, focal audio, or soundscape audio.

The design yields four practical properties that the heavier community pipelines give up. It is *class-extensible*: the class set is stored as a data artifact, so introducing or removing a species means creating or dropping its prototype centroids rather than retraining a head, and unlike the supervised SSM and SED models, the model never bakes a fixed class count into its parameters. It is *inspectable*: because each class is scored by similarity to concrete exemplars, a prediction can be traced to the exemplar it most resembled ("most similar to this example"), which both explains the decision and gives a direct way to audit and curate the bank, for instance to identify and prune false-positive exemplars. It is *cheap*: with no ensemble there is no model-weighting machinery to manage, and the model fits in seconds

to minutes on CPU rather than hours on GPU. And it is *portable*: every component was chosen to generalize across sites rather than to exploit the competition metric.

Self-training on the unlabeled soundscape pool [22] was the single largest performance step, but only for one round. Our most useful deployment finding is asymmetric: a heavy ensemble *teacher* produces better pseudo-labels, so one can fit it once on the small labeled set, use it only to label the pool, and then deploy the single lightweight model, keeping most of its gain without ever serving it. The per-class environmental head is a cheap further lift, but it is per-class and site-correlated, so we report it as an extension, not the headline.

The single agile model reaches 0.917 public / 0.923 private macro ROC–AUC (0.931 private with the environmental extension), within roughly 0.04 private of the winning entry among 4,091 teams, at a small fraction of the training and inference cost: evidence that, in this focal-to-soundscape transfer setting, most of the macro ROC–AUC of a heavy supervised ensemble can be recovered by a model that is simple, efficient, and straightforward to extend to new species and new sites.

Acknowledgments

The four-model ensemble reported here as an upper bound incorporates two components reproduced from the public BirdCLEF Kaggle community: the SED model follows a notebook by Tucker Arrants [39] and the SSM model a notebook by Vyanktesh Dwivedi [38]. We thank them and the wider BirdCLEF Kaggle community, whose openly shared notebooks and discussions these models build on.

Declaration on Generative AI

During the preparation of this work, the author used Anthropic Claude Opus 4.7 and 4.8 in order to: Automate experiments and submissions, paraphrase and reword text. Experiments were all designed and orchestrated by the author. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

A. Dataset characterization

This appendix collects the per-taxon, genus, site/time, and taxon–site tables summarized in Section 3; all counts derive from the canonical partition of Table 1.

A.1. Per-taxon regime structure

The partition is highly uneven across taxa (Table 10). The no-soundscape species are overwhelmingly avian: **134 of the 159** out-of-domain classes (89%) are Aves, so the bulk of the taxonomy (and plausibly the hidden test) is in-vocabulary but out-of-domain. The subset of soundscape-labeled classes with enough positives to fit on directly (≥ 5 labeled windows) is small for every taxon except Amphibia and Insecta: only 15 Aves, 3 Mammalia, and a single Reptilia clear the floor. Any *per-taxon* fitting is therefore data-starved for three of the five taxa and must pool or shrink across regimes rather than estimate each taxon independently.

A.2. Genus structure and cross-regime sharing

The species rows are genus-*sparse*: most genera contribute a single species, which limits how much a genus-level prior can pool (Table 11). Crucially, genus also provides almost no bridge *between* the labeled in-domain set and the no-soundscape species. Only **13%** of out-of-domain species (21 of 159) have a labeled congener (Table 12), and within the 134-class avian bulk that figure falls to 6%. The one exception is Amphibia, where 72% of out-of-domain species share a genus with a labeled relative and genus structure is dense.

Table 10
Per-taxon regime breakdown (regime cells as defined in Section 3.2). Parenthetical figures count classes with ≥ 5 labeled soundscape windows (i.e. fit-able on the real soundscape surface). Column totals recover the canonical 2 × 2: IV+ID 50, IV+OOD 159, OOV+ID 25.

Taxon	IV+ID (fit ≥ 5)	IV+OOD	OOV+ID (fit ≥ 5)	Total
Aves	28 (15)	**134**	0	162
Amphibia	17 (16)	18	0	35
Insecta	0	3	**25 (24)**	28
Mammalia	4 (3)	4	0	8
Reptilia	1 (1)	0	0	1
Total	50 (35)	159	25 (24)	234

Table 11
Genus composition per regime cell (Section 3.2). The species are mostly one-per-genus, bounding genus-level pooling; the insect sonotypes carry no species-level taxonomy.

Cell	Species	Genera	Multi-species genera
IV+ID	50	43	6 (*Leptodactylus* ×3; *Physalaemus*, *Scinax*, *Dendropsophus*, *Amazona*, *Cyanocorax* ×2)
IV+OOD	159	128	20 (*Leptodactylus* ×6, *Synallaxis* ×5, *Physalaemus*/*Thamnophilus*/*Elaenia* ×3, …)
OOV+ID	25	—	none (the 25 insect sonotypes have no genus structure)

Table 12
Cross-regime genus sharing: out-of-domain species whose genus also appears in the labeled in-domain set, the only genus-level ID→OOD transfer hook. Dense only for Amphibia.

Taxon	OOD species with an ID congener
Amphibia	**13 / 18 (72%)**
Aves	8 / 134 (6%)
Insecta	0 / 3
Mammalia	0 / 4
Overall	**21 / 159 (13%)**

A.3. Site, time, and taxon–site confounding

Site and time coverage of the labeled versus unlabeled soundscapes is given in Table 13. Beyond the marginal coverage summarized in Section 3.6, the per-window species count is strongly structured by site and hour: the per-site mean ranges from 1.05 (S09) to 5.37 (S08), a 5.1× spread; hour-of-day alone explains $R^2 \approx 0.25$ within the one multi-hour site (S22); and a two-way decomposition attributes ≈42% of the density variance to site and hour jointly (site-unique 0.21, hour-unique 0.16, shared 0.04). We nonetheless keep the admission density target *global* (4.22): a per-site target would have to be estimated from per-site labeled soundscapes, but the sites that dominate the unlabeled pool have none (95% of the pool comes from four sites, and the two largest non-S22 sites, S01 and S02, 46% of the pool, carry zero labeled soundscapes), and the only labeled site large enough to set its own target is S22, whose 4.34 mean is already close to the global 4.22.

A direct contingency analysis of the 739 labeled windows shows that taxon is almost perfectly confounded with site (Table 14): each labeled site captured essentially one taxonomic community. S22, which supplies 65% of the labels, is 98% Amphibia and 0% Insecta; insects appear only at S08/S15/S19/S23;

Table 13
Site and time coverage of the labeled versus unlabeled soundscapes. Hour-of-day is UTC; the data is night/crepuscular with near-zero midday coverage.

	Sites	Top sites (window share)	Hour-of-day (UTC)	Months (years)
Labeled (739 w)	9	S22 65%, S08 8%, S15 6%, S19/S23 5%	~90% in 18–06 h, ~0% midday	10 (2021–2025)
Unlabeled (127k w)	21	S22 32%, S02 24%, S01 22%, S13 18%	~98% in 18–06 h, ~0% midday	46 (2014–2025)

Table 14
Per-site taxon composition of the 739 labeled windows: the share of each site's windows containing at least one species of the taxon (%). Taxon is nearly confounded with site, so pooled taxon×environment effects are site×environment artifacts.

Site	Windows	Aves	Amphibia	Insecta	Mammalia	Reptilia
S22	477	25	**98**	0	3	0
S08	60	28	0	**100**	0	17
S15	48	**100**	0	**100**	0	0
S19	36	3	**100**	67	0	0
S23	36	61	0	**100**	33	8
S03	24	0	**100**	0	0	0
S13	24	0	**100**	0	0	0
S09	19	37	0	0	**68**	0
S18	15	0	**100**	0	0	0

mammals concentrate at S09. Consequently any taxon×environment pattern read off the *pooled* labels is really a site×environment artifact. Restricting to S22 (the only site large enough to control for) collapses every pooled pattern except **Aves** × **diel**: Aves is night-enriched (presence 0.41 vs 0.25, $\log_2$ +0.75, Fisher $p < 0.001$), while Insecta and Reptilia are unmeasurable (absent from S22) and Amphibia saturates (present in nearly every S22 cell, leaving no contrast). This is why the environmental head (Section 4.6) estimates its profiles from the *unlabeled* pool (taxon-mixed across 21–23 sites) rather than the site-confounded labels, and why only its Aves × diel interaction is corroborated by the labeled data while its Amphibia × rain prior rests on external ecology.

B. Additional ablations

These controls corroborate the design choices of Section 5.3 and the deployed scorer's interpretability. The bank- and embedder-level screens are each a single-seed leaderboard run that exhibits the offline↔leaderboard inversion which governs our selection discipline; Table 17 additionally reports the per-species Perch composite referenced in Section 5.2.1.

B.1. Fitted global-LR coefficients

Table 15 lists the deployed env-OFF primary scorer's standardized coefficients (mean ± std over the five seed-fold scorers). The six prototype-cosine `proto_*` order statistics carry essentially all the weight; the de-biased `masked_logit` adds a small but very stable positive nudge (+1.15, std 0.02); and the entire Gaussian-density family together with the cohort z-scores is inert (≈ 0) under the per-feature standardization, so roughly half of the thirteen-feature vector is dead weight. This is the quantitative basis for the reading that *distance*, not density, is the operative signal, and it makes the model's decisions legible feature by feature.

Table 15
Deployed env-OFF primary global-LR coefficients (standardized; mean ± std over five seed-fold scorers), sorted by magnitude. The Gaussian-density and cohort-z features are inert.

Feature	Coefficient
`proto_mean`	6.14 ± 1.09
`proto_std`	5.59 ± 2.35
`proto_max`	4.48 ± 0.94
`proto_margin`	4.39 ± 0.76
`proto_max_minus_top2`	-2.84 ± 0.74
`proto_max_other`	-1.74 ± 0.15
`masked_logit` (de-biased)	1.15 ± 0.02
`z_pmax`	-0.05 ± 0.04
`gauss_ll_max_other`	≈ 0
`gauss_ll`, `gauss_margin`, `gauss_mahalanobis`, `z_gll`	≈ 0 (dead)
intercept	2.37 ± 0.93

Table 16
Pseudo-weight sweep (five seeds, single-model OOF-isolated teacher). Monotone on both surfaces; full weight is the iter-1 operating point.

Pseudo-weight	LB public	LB private
0.3	0.9008 ± 0.0038	0.9103 ± 0.0034
0.5	0.9032 ± 0.0037	0.9131 ± 0.0033
1.0	**0.9060 ± 0.0034**	**0.9158 ± 0.0033**

Table 17
Composite routings, five seeds: scoring the in-vocabulary, soundscape-labeled species by the zero-shot logit versus a 16-shot in-domain probe. The two routings differ only on these species, isolating logit-versus-probe; the probe wins by +0.0031 public.

Scorer	LB public	LB private
16-shot in-domain probe	**0.8927 ± 0.0044**	0.8999 ± 0.0035
Zero-shot masked logit	0.8896 ± 0.0046	0.8943 ± 0.0044

B.2. Pseudo-weight sweep

Table 16 sweeps the weight on admitted pseudo-labeled rows (ground-truth rows fixed, single-model OOF-isolated teacher); it is monotone on both surfaces with no per-seed sign flips, so full weight (1.0) is the iter-1 operating point (Section 5.4.1).

B.3. Per-species Perch composite

Isolating logit-versus-probe on the in-vocabulary, soundscape-labeled species (the two routings differ only on these), the 16-shot in-domain probe beats the zero-shot logit by +0.0031 public (Table 17). The proper comparison against GREBE, holding the no-soundscape routing fixed, is the per-bucket isolation of Section 5.2.1 (Table 5).

B.4. Prototype-bank mixup operator

How a focal clip is combined with a soundscape background to form a no-soundscape prototype is the operator that bridges the domain gap. We compare element-wise max, convex linear, physical waveform mixing (before the embedder), and a raw no-mixup key, all else fixed (Table 18). The leaderboard ranks element-wise max first, with *raw* (no mixup) second, ahead of both convex and physical mixup, so only

Table 18
Prototype-key mixup operator (single seed). Element-wise max wins; raw no-mixup beats both convex and physical mixup. Offline labeled ranks them in the reverse order.

Operator	Offline labeled	LB public	LB private
Element-wise max	0.9018	**0.9005**	**0.9021**
Raw (no mixup)	0.8887	0.8919	0.9029
Convex embedding ($\lambda \sim$ Beta)	0.9084	0.8595	0.8448
Physical waveform overlap	0.9119	0.8492	0.8286

Table 19
Embedder views (single seed). More views are monotonically worse on the leaderboard, exactly inverting the offline labeled ranking.

Configuration	Offline labeled	LB public	LB private
Perch (single view)	0.8996	**0.9005**	**0.9021**
+ max-pool	0.9065	0.8946	0.8996
+ BirdNET	0.9067	0.8878	0.8920
+ both	0.9091	0.8809	0.8888

element-wise max adds value over doing no focal×soundscape mixup at all. The offline labeled ranking is the exact reverse (physical waveform mixup, best offline, is the worst deployed): element-wise max preserves a strong, clean species signature in the no-soundscape prototypes that dominate the hidden test, while convex and physical blends attenuate the focal signal into the background. Had we gated on the offline number we would have shipped the worst operator.

B.5. Embedder views: extra views help offline but hurt on the leaderboard

Starting from the base Perch-mean view, we add a spatial max-pool view, a BirdNET-v2.4 view [5], and both, with the fold partition held bit-identical across configurations. We never concatenate the raw embeddings; instead each view contributes its own twelve-dimensional block of class-relative dissimilarity features to the same global scorer. Table 19 shows a clean offline↔leaderboard inversion: the leak-prone labeled surface ranks the extra views as mildly additive, while the leaderboard ranks them monotonically *worse*. Because the offline surface over-credits the added capacity, we deploy the single Perch-mean view and do not use the multi-view variants.

References

[1] S. Kahl, T. Denton, L. Sugai, L. Piatti, W. H. Arruda Moreno, M. M. Barbosa, M. Eibl, C. Z. Fieker, C. M. Garcia, D. L. Hokama Sousa, J. E. d. A. Júnior, R. C. Kridler, M. Lasseck, A. V. d. Melo, H. Müller, M. d. O. Neves, L. K. Reis, Matheus Gonçalves dosand Sampaio Teles, K.-L. Schuchmann, J. L. M. M. Sugai, P. d. N. Azevedo, Kirk Thiago Pedrosoand Lopes, M. I. Marques, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF+ 2026: Acoustic species identification in the Pantanal, South America, in: Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, 2026.

[2] L. Picek, L. Adam, S. Kahl, R. Bossy, L. Chrobak, H. Goëau, K. Papafitsoros, H. Klinck, W.-P. Vellinga, R. Planqué, T. Denton, K. Barnard, C. Nédellec, L. Deléger, M. Courtin, G. Martellucci, I. Moummad, F. Vinatier, P. Bonnet, A. Joly, Overview of LifeCLEF 2026: Ai challenges for biodiversity understanding and ecosystem management, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2026.

[3] S. Kahl, T. Denton, H. Klinck, V. Ramesh, V. Joshi, M. Srivathsa, A. Anand, C. Arvind, H. Cp, S. Sawant, et al., Overview of BirdCLEF 2024: Acoustic identification of under-studied bird species

in the Western Ghats, in: Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum, CEUR-WS, 2024, pp. 1948–1957.
[4] J. S. Cañas, S. Kahl, T. Denton, M. P. Toro-Gómez, S. Rodriguez-Buritica, J. L. Benavides-Lopez, J. S. Ulloa, P. Caycedo-Rosales, H. Klinck, H. Glotin, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF+ 2025: Multi-taxonomic sound identification in the middle magdalena valley, colombia, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, 2025.
[5] S. Kahl, C. M. Wood, M. Eibl, H. Klinck, BirdNET: A deep learning solution for avian diversity monitoring, Ecological Informatics 61 (2021) 101236.
[6] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, M. D. Plumbley, PANNs: Large-scale pretrained audio neural networks for audio pattern recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020) 2880–2894.
[7] V. Sydorskyi, F. Gonçalves, Tackling domain shift in bird audio classification via transfer learning and semi-supervised distillation: A case study on BirdCLEF+ 2025, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, 2025. Code: https://github.com/VSydorskyy/BirdCLEF_2025_2nd_place.
[8] B. van Merriënboer, V. Dumoulin, J. Hamer, L. Harrell, A. Burns, T. Denton, Perch 2.0: The bittern lesson for bioacoustics, 2025. arXiv:2508.04665.
[9] B. Ghani, T. Denton, S. Kahl, H. Klinck, Global birdsong embeddings enable superior transfer learning for bioacoustic classification, Scientific Reports 13 (2023) 22876.
[10] A. Burns, L. Harrell, B. van Merriënboer, V. Dumoulin, J. Hamer, T. Denton, Perch 2.0 transfers 'whale' to underwater tasks, 2025. arXiv:2512.03219, neurIPS 2025 Workshop: AI for Non-Human Animal Communication.
[11] L. Rauch, R. Schwinger, M. Wirth, R. Heinrich, D. Huseljic, M. Herde, J. Lange, S. Kahl, B. Sick, S. Tomforde, C. Scholz, BirdSet: A large-scale dataset for audio classification in avian bioacoustics, 2024. arXiv:2403.10380.
[12] T. Mensink, J. Verbeek, F. Perronnin, G. Csurka, Distance-based image classification: Generalizing to new classes at near-zero cost, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 2624–2637.
[13] J. Snell, K. Swersky, R. S. Zemel, Prototypical networks for few-shot learning, 2017. arXiv:1703.05175, published at NeurIPS 2017.
[14] B. Williams, B. van Merriënboer, V. Dumoulin, J. Hamer, L. Harrell, T. Denton, et al., Leveraging tropical reef, bird and unrelated sounds for superior transfer learning in marine bioacoustics, 2024. arXiv:2404.16436, published in Philosophical Transactions of the Royal Society B, 2025.
[15] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, 2018. arXiv:1804.09458, published at CVPR 2018.
[16] E. Pękalska, R. P. W. Duin, The Dissimilarity Representation for Pattern Recognition: Foundations and Applications, volume 64 of *Series in Machine Perception and Artificial Intelligence*, World Scientific, 2005.
[17] Y. M. G. Costa, D. Bertolini, A. S. Britto, G. D. C. Cavalcanti, L. E. S. Oliveira, The dissimilarity approach: a review, Artificial Intelligence Review 53 (2019) 2783–2808.
[18] L. Nanni, S. Brahnam, A. Lumini, G. Maguolo, Animal sound classification using dissimilarity spaces, Applied Sciences 10 (2020) 8578.
[19] R. Kurinchi-Vendhan, S. Beery, Finding needles in the haystack: Transductive active labeling in ecology, 2026. URL: https://arxiv.org/abs/2606.03821. arXiv:2606.03821.
[20] Y. Wang, W.-L. Chao, K. Q. Weinberger, L. van der Maaten, Simpleshot: Revisiting nearest-neighbor classification for few-shot learning, https://arxiv.org/abs/1911.04623, 2019. ArXiv:1911.04623.
[21] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, H. Li, Tip-adapter: Training-free adaption of CLIP for few-shot classification, in: European Conference on Computer Vision (ECCV), 2022.
[22] D.-H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: ICML Workshop on Challenges in Representation Learning, 2013.
[23] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, T. Shinozaki, FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling, 2021. arXiv:2110.08263, published

at NeurIPS 2021.
[24] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, X. Xie, FreeMatch: Self-adaptive thresholding for semi-supervised learning, 2023. arXiv:2205.07246, published at ICLR 2023.
[25] A. Miyaguchi, A. Cheung, M. Gustineli, A. Kim, Transfer learning with pseudo multi-label birdcall classification for DS@GT BirdCLEF 2024, in: Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum, 2024.
[26] V. Dumoulin, O. Stretcu, J. Hamer, L. Harrell, R. Laber, H. Larochelle, B. van Merriënboer, T. Denton, et al., The search for squawk: Agile modeling in bioacoustics, 2025. arXiv:2505.03071.
[27] W. J. Scheirer, A. De Rezende Rocha, A. Sapkota, T. E. Boult, Toward Open Set Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 1757–1772. URL: http://ieeexplore.ieee.org/document/6365193/. doi:10.1109/TPAMI.2012.256.
[28] S. Vaze, K. Han, A. Vedaldi, A. Zisserman, Open-Set Recognition: A Good Closed-Set Classifier is All You Need?, ArXiv abs/2110.06207 (2021). URL: https://api.semanticscholar.org/CorpusID:238634102.
[29] A. Miyaguchi, M. Gustineli, A. Cheung, Distilling spectrograms into tokens: Fast and lightweight bioacoustic classification for BirdCLEF+ 2025, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, 2025. Code: https://github.com/dsgt-arc/birdclef-2025.
[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems (NeurIPS), 2019.
[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
[32] ONNX Runtime developers, Onnx runtime, https://onnxruntime.ai/, 2021. Version 1.21.0.
[33] R. Auckenthaler, M. Carey, H. Lloyd-Thomas, Score normalization for text-independent speaker verification systems, Digital Signal Processing 10 (2000) 42–54.
[34] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. Diez Sánchez, J. Černocký, Analysis of score normalization in multilingual speaker recognition, in: Interspeech 2017, 2017, pp. 1567–1571.
[35] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, 2023. arXiv:2312.00752.
[36] A. Gu, K. Goel, C. Ré, Efficiently modeling long sequences with structured state spaces, 2022. arXiv:2111.00396, published at ICLR 2022.
[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762, published at NeurIPS 2017.
[38] V. Dwivedi, BirdCLEF 2026: ONNX Perch sequence modeling, https://www.kaggle.com/code/vyankteshdwivedi/birdclef-2026-onnx-perch-sequence-modeling, 2026. Kaggle notebook, accessed 2026-06-17.
[39] T. Arrants, BC2026 distilled SED, https://www.kaggle.com/code/tuckerarrants/bc2026-distilled-sed?scriptVersionId=315112860, 2026. Kaggle notebook (version 315112860); described in discussion https://www.kaggle.com/competitions/birdclef-2026/discussion/694479, accessed 2026-06-17.
[40] J. W. Spencer, Fourier series representation of the position of the sun, Search 2 (1971) 172.
[41] P. Zippenfenig, Open-meteo.com weather API, 2023. URL: https://open-meteo.com/. doi:10.5281/zenodo.7970649.
[42] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in: Advances in Neural Information Processing Systems (NeurIPS), 2017.