Leveraging Public Resources as a Late Entrant in BirdCLEF+ 2026: An Empirical Study of Embedding-Space Ensemble Diversity and Its Limits

Edidiong-Abasi Ita Anwanane[1]

[1]*Independent Researcher, Lagos, Nigeria*

Abstract

In this paper, my late-entry to the BirdCLEF+ 2026 multi-taxon passive-acoustic monitoring challenge is reported and analyzed and it is used as a case study to examine the limits of the leverage provided by public resources in tightly clustered Kaggle competitions. Working with roughly fifteen days before the deadline and only freely available compute, I moved from a from-scratch baseline (public ROC-AUC 0.787) to a competitive public-pipeline fork (0.949), then attempted to exceed that ceiling with a compact multilayer-perceptron head trained on Perch v2 embeddings newly extracted for all 10,658 unlabelled training soundscapes (127,896 windows; the public cache covers only 59 files). Despite achieving healthy standalone validation and moderate output decorrelation or uncoupling from the underlying Perch logits, three blend configurations of the head produce monotone score declines and a pre-registered probe of the blend ratio is exactly neutral. I interpreted this as signal redundancy: a second classifier on the same foundation-model embeddings, however architecturally distinct, adds noise rather than complementary signal to an already saturated ensemble. I also document that hundreds of teams share the identical 0.949 public score, and trace its dispersal on the private leaderboard, where my submission scored 0.94063 (rank 1,313 of 4,092, top 32.1%), climbing 119 places. The lesson for late entrants is that once the public ceiling is reached, the productive path is not to chase it but to invest in a truly orthogonal signal; I document a reproducible methodology for extending the public Perch v2 embedding cache to the full unlabelled soundscape set as an enabling step for that work.

Keywords

bioacoustic classification, passive acoustic monitoring, Perch v2, BirdCLEF, transfer learning, pseudo-labelling, ensemble diversity, negative results, Kaggle competition methodology

1. Introduction

BirdCLEF+ 2026 [1] requires that its participants identify wildlife species (birds, amphibians, mammals, reptiles, and insects) from five-second windows of continuous soundscape audio recorded by an increasing network of acoustic recorders deployed throughout the Brazilian Pantanal. The task is central to scalable biodiversity monitoring: the volume of audio collected by passive recorders has long exceeded any human capacity to review it manually, and reliable automated identification is a prerequisite for using such data in conservation decisions. The 2026 edition spans 234 target classes (many of which are rare, several of which are non-avian, and a substantial fraction of the insects represented only as unidentified sonotypes) and is evaluated by macro-averaged ROC-AUC, weighting every present species equally regardless of its frequency. The task runs as part of the LifeCLEF 2026 lab [2], which hosts a suite of biodiversity-identification challenges across taxa and modalities.

Across recent editions of the BirdCLEF series, public sharing of pre-trained models, embedding caches, distilled student models, and tuned inference pipelines has progressively smoothed the competitive landscape [3, 4]. The current edition is no exception: freely available foundation-model embeddings from Perch v2 [5], distilled sound-event-detection models, and a carefully tuned multi-component inference pipeline together produce a strong public baseline that any competitor can adopt with a single notebook fork. By the deadline for final submission, this baseline scored 0.949 on the public leaderboard. This three-decimal place score was shared by hundreds of the 4,092 participating teams.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

didianwanane@gmail.com (E. I. Anwanane)

This compression has a specific consequence for late entrants. With limited time and limited compute resources, the strategically rational first step is to fork the public pipeline. This is a step that closes most of the gap between the entrant and the top leaderboard placers at essentially zero cost. But once inside the resulting tied cluster, the entrant faces a sharply different problem: the public recipe is exhausted, and any further improvement requires a truly complementary signal that the recipe does not already provide. Most of the existing BirdCLEF working notes have been written by teams that competed for the entire duration of the challenge and present elaborate winning pipelines. The complementary question, which is what a disciplined late entrant can and cannot achieve from inside the cluster, has not been extensively addressed to my knowledge.

This paper makes three contributions:

- **Strategic framing.** I characterise the late-entry condition empirically, quantifying both the value of the leverage provided by public resource(s) (a measured $+0.162$ jump in macro-averaged ROC-AUC from a from-scratch baseline to a tuned public pipeline) and the size of the resulting tied cluster (hundreds of teams at the identical public score of 0.949) together with its dispersal on the private leaderboard.
- **A controlled negative result.** I test the most natural follow-up hypothesis for a late entrant, which is that a custom classifier head trained on the same foundation-model embeddings the public pipeline already uses can provide ensemble diversity, and report a clean monotone failure across three blend configurations, including a class-specific variant restricted to the head's strongest taxon. Despite a standalone validation macro-AUC of 0.852 and a healthy median Spearman correlation of 0.58 between the head's predictions and the raw Perch logits, no positive blend weight matches or exceeds the baseline. A mechanistic interpretation is provided: the sequence model already in the pipeline consumes the same embeddings, and the head's apparent diversity resides in noise rather than complementary signal.
- **A reproducible enabling methodology.** As an enabling step for the above experiment, I extracted Perch v2 embeddings for all 10,658 unlabelled training soundscapes (127,896 windows, 1,536-dimensional embeddings plus mapped 234-class logits). The existing public embedding cache covers only the 59 fully labelled soundscape files. Section 3.3 documents the extraction methodology in sufficient detail to be reproduced by other researchers. The resulting artefact is reserved for a planned follow-up study; public release will be considered alongside that work, consistent with the competition's CC BY-NC-SA data-licence terms.

These contributions are framed to be useful both to other late entrants and to researchers thinking about mature, foundation-model-driven competitions, where the central practical lesson is that once the public ceiling is reached, any additional leaderboard tuning has a non-positive expected value and only a very orthogonal signal moves the result upward. Section 2 places the work in context, Section 3 describes the methods, Section 4 reports the seven submissions at the empirical core of the study, and Section 5 interprets the negative result and the cluster phenomenon.

2. Related Work and Background

2.1. The BirdCLEF Series and Passive-Acoustic Monitoring

Passive acoustic monitoring (PAM) using autonomous recorders has emerged as a scalable instrument for assessing biodiversity, particularly in remote or fragile ecosystems where traditional field surveys are expensive and logistically difficult [6]. The BirdCLEF series [3, 4] has, over successive editions, framed PAM as a machine-learning challenge: identify which species are present from short windows of continuous soundscape recording. Recent editions have progressively broadened both the taxonomic scope (BirdCLEF+ 2026 includes amphibians, mammals, reptiles, and insect sonotypes alongside birds) and the geographic and ecological diversity (the 2026 data is drawn from the Pantanal wetlands of Brazil).

The progression of winning approaches across editions tracks the wider field. Earlier editions were dominated by convolutional ensembles that were spectrogram based. More recent editions have been dominated by transfer from large bioacoustic foundation models namely BirdNET [7] and, in the current edition, Perch v2 [5] either as fixed feature extractors or via fine-tuned heads, frequently combined with self-training on the abundant unlabelled soundscape audio.

2.2. Public Resources and the BirdCLEF+ 2026 Public Pipeline

A distinctive feature of recent BirdCLEF editions is the extent of public sharing on the Kaggle platform. For BirdCLEF+ 2026, the public ecosystem includes: Perch v2 ONNX exports tuned for the competition's CPU-only inference budget; distilled sound-event-detection (SED) student models trained on Perch outputs; embedding caches over the labeled soundscape subset; and a family of inference notebooks combining these components with a prototype state-space sequence model (ProtoSSM), Bayesian site-by-hour priors, isotonic calibration, and a five-gate rank-percentile blend. By the time of my entry, several tuned variants of this pipeline, which were released as a numbered sequence of incrementally refined public notebooks (of which the variant labelled "exp019" was the highest scoring publicly available at the time), had converged on public scores in the 0.948 to 0.949 range. I refer to this collectively as the public Perch+ProtoSSM+SED pipeline. I acknowledge the authors of these resources in Section 6 and identify the specific notebooks I built on them in Section 3.

2.3. Ensemble Diversity and Foundation-Model Heads

Classical results on ensemble learning show that combining classifiers improves performance to the extent that the classifiers' errors are uncorrelated [8, 9]. In foundation-model-based pipelines, this poses a specific challenge: when multiple downstream components consume the same upstream representation, their errors are often more correlated than their architectures would suggest, because the shared representation has already filtered the input space in a particular way. The negative result reported here is an instance of this phenomenon, observed in a controlled setting where the candidate ensemble component looked promising on every standalone metric I could measure.

2.4. Negative Results and Late-Entry Methodology

Working notes from competitions are, by selection, biased toward winning approaches: teams that found something that worked describe what they did; teams whose hypotheses did not pan out frequently do not write up. At the community level, the cost of this is that the same dead ends are rediscovered repeatedly. Where documented negative results can be found in this literature, they tend to be brief asides within winning-approach papers rather than systematic studies. This paper seeks to address this gap on a single, well-motivated question, in the specific context of a late entrant who cannot afford open-ended exploration.

3. Methodology

3.1. Task, Data, and Evaluation

I follow the official competition setup [1]. The training data comprise:

- focal recordings of individual species drawn from xeno-canto and iNaturalist, totalling 35,549 files spread across the 234 target classes (with strongly imbalanced per-class counts and Xeno-canto quality ratings); and
- 10,658 one-minute Pantanal soundscape files, of which a small subset (59 files) is fully annotated by experts at a five-second resolution. All audio is resampled to 32 kHz mono. The hidden test set consists of approximately 600 one-minute soundscape files recorded at sites and dates that do not overlap with the labelled training portion. Submissions are scored by macro-averaged ROC-AUC over the subset of the 234 classes that have at least one positive label in the test data.

BirdCLEF+ 2026 is a code competition: submissions are Kaggle notebooks that the platform re-executes against the hidden test data under a 90-minute CPU-only runtime budget with internet access disabled. This constraint shaped my design throughout: any candidate ensemble component had to run within the remaining budget after the public pipeline's existing components.

3.2. Public Baseline Pipeline

My baseline is a direct fork of the public Perch+ProtoSSM+SED pipeline introduced in Section 2, specifically the "exp019" variant by Derek (@sunderekkiz), which was the highest scoring publicly available variant of the family at the time of my entry. Briefly, the pipeline at inference time:

- loads each 60-second test file and slices it into twelve 5-second windows;
- runs each window through Perch v2 (via an ONNX export tuned for CPU) to obtain a 1,536-dimensional embedding and a vector of raw Perch logits which are mapped to the 234 competition classes via scientific name matching plus genus-level proxy mapping for unmatched Aves/Amphibia/Insecta species (with 31 classes, the 25 insect sonotypes plus six non-bird vertebrates whose scientific names do not match any entry in the Perch label set, receiving no Perch signal at all);
- trains, in-notebook, a prototype state-space sequence model (ProtoSSM) on the 708 fully labelled soundscape windows, which is augmented with joint site-by-hour Bayesian priors estimated from the training data, MLP probes on the Perch embeddings with PCA compression, and a ResidualSSM correction trained against the ProtoSSM's predictions;
- infers with a five fold distilled SED student ensemble on mel-spectrograms of the same audio; and
- combines the ProtoSSM and SED predictions through a rank-percentile blend (default weights 0.60 ProtoSSM / 0.40 SED), followed by a five-gate post-processing chain (noise suppression, temporal continuity smoothing, SED-spike preservation, sonotype mirroring, and adaptive per-class thresholding).

The full pipeline runs in approximately 41 minutes on Kaggle's CPU-only submission infrastructure. I did not modify this pipeline structurally; my integration adds a single ensemble term that I describe in Section 3.4.

3.3. Embedding Extraction over Unlabelled Soundscapes

The existing public embedding cache covers only the 59 fully labelled soundscape files (708 windows). To train a head on a meaningful amount of in-domain audio, I extracted Perch v2 embeddings and mapped logits for all 10,658 training soundscapes (127,896 five-second windows) using the same ONNX model and the same class-mapping logic as the baseline pipeline, ensuring that the resulting embeddings are interchangeable with the cache that ProtoSSM consumes internally. The extraction was performed in a single Kaggle session on a free T4 GPU and took 264.7 minutes (approximately 4.4 hours) of wall-clock time. The resulting artefact comprises an 826 MB compressed NumPy archive (embeddings and mapped logits) and a small Parquet file of per-window metadata (row identifier, file name, recording site, and recording hour in UTC).

3.4. Head Architecture and Training

I trained a compact multilayer perceptron mapping Perch embeddings to 234 class species predictions. The architecture is a two-hidden-layer MLP with hidden dimension of 512, layer normalisation, GELU activations, and dropout of 0.3:

$$1536 \to [\text{Linear, LayerNorm, GELU, Dropout}] \to 512 \to [\text{Linear, LayerNorm, GELU, Dropout}] \to 512 \to \text{Linear} \to 234.$$

The model has approximately 1.3 million parameters and serialises to roughly 5MB. Inference for the full hidden test set (≈7,200 windows) completes in seconds on CPU.

The training combined two sources. First, I used the 708 fully labelled soundscape windows with their multi-label targets that are human-annotated. Secondly, I generated pseudo-labels for the 127,896 unlabelled windows by thresholding the Perch logits. After preliminary experiments were conducted with a single global threshold (which produced an implausible mean of 48 positive labels per window), I adopted a more careful scheme: per-class thresholds were set at the 97th percentile of each class's logit distribution (with a minimum floor of 1.5 to suppress noise positives in classes with low overall logit ranges), followed by a hard cap of five positive labels per window. The 44 classes for which Perch produces no usable signal on the soundscape data (defined as a maximum logit below 1.0 across all 127,896 windows, a superset of the 31 classes unmappable by name, and additionally includes classes that map by name but never activate confidently on this data) received no pseudo-labels; those classes are learnable only from the 708 real-labelled windows. After the top-K cap, the resulting pseudo-labelled training set averages roughly 3.7 positive labels per window, in line with what an annotator might mark for a 5-second wilderness clip.

The training used a weighted multi-label binary cross-entropy loss with sample weights 1.0 on real-labelled windows and 0.3 on pseudo-labelled windows. I optimised with AdamW (learning rate 10^{-3}, weight decay 10^{-4}) and a cosine learning rate schedule over 25 epochs, with batch size 1024 and gradient clipping at norm 5. I used GroupKFold with five folds, grouping windows by their parent soundscape filename to prevent within-file leakage, and trained the head reported in this paper on fold 0. Validation was restricted to the real-labelled windows in the held-out fold (144 windows spanning 37 of the 234 classes); the head's macro-ROC-AUC on this set reached 0.852 at epoch 17, after which validation performance plateaued.

3.5. Ensemble Integration

I inserted the trained head into the baseline pipeline as a third term in the rank-percentile blend, placed immediately before the existing five-gate post-processing chain (so that downstream noise-suppression, temporal-continuity smoothing, and other gates operate on the new blended predictions exactly as before). At inference time, the head reuses the Perch embeddings that the baseline pipeline has already computed for the test set, while contributing only seconds of additional inference. I computed the head's predicted probabilities, and aligned them to the row ordering of the ProtoSSM and SED outputs, rank-transform them per class, and blend:

$$\text{pred} = (1 - \alpha) \cdot \text{pred}_{\text{two-way}} + \alpha \cdot \text{rank}(\text{head\_probs}) \tag{1}$$

where $\text{pred}_{\text{two-way}}$ is the existing ProtoSSM-plus-SED rank blend and α is the head's contribution weight. I also implemented a class-specific variant in which the head is blended only on a subset of class columns (specifically the 162 Aves classes), leaving non-Aves columns as the pure two-way blend; this targets the taxon for which both the head's standalone validation and Perch's underlying representation are strongest.

3.6. Submission Selection and Hedging

BirdCLEF+ 2026 allows two final submissions, with the better scoring of the two on the private leaderboard that determines the entrant's final standing. Because the public score was tied at 0.949 with hundreds of other teams (entrants), I expected the private leaderboard to redistribute this tie based on small inter-pipeline differences not visible at the public three-decimal scoring. Rather than select two identical submissions, I chose two that produce different prediction files. These were the tuned baseline (0.60/0.40 blend) and a blend-ratio variant (0.55/0.45), with the goal being to draw twice from the private-test distribution. Both submissions scored 0.949 on the public set, so this hedge carried no public-score cost while providing protection against shake-up variance after submissions. As reported in Section 4.5, the redistribution did occur post-submission and the chosen submissions climbed 119 places relative to their final public position.

#	Configuration	Public LB	Role in study
1	From-scratch EfficientNet-B0 (single fold)	0.787	Solo-effort anchor
2	Public Perch v2 + ProtoSSM + SED fork	0.944	First public-resource leverage
3	exp019 fork (tuned public pipeline)	0.949	Baseline; final submission A
4	exp019 + custom head, uniform $w = 0.07$	0.947	Head ablation
5	exp019 + custom head, uniform $w = 0.03$	0.948	Head weight sweep
6	exp019 + custom head, Aves-only $w = 0.07$	0.948	Class-specific head variant
7	exp019 with blend ratio 0.55/0.45 (no head)	0.949	Blend test; final submission B

Table 1: Chronological summary of all scored submissions.

3.7. Reproducibility

The baseline pipeline is a public Kaggle notebook (acknowledged in Section 6), and all components it depends on (the Perch v2 ONNX export, the distilled SED ensemble, the labelled-soundscape embedding cache, and the fold definitions) are also public Kaggle datasets cited there with stable links. The contributions specific to the author are reported here, namely the embedding extraction over the full unlabelled soundscape set (Section 3.3), the pseudo-label generation scheme (Section 3.4, Appendix Table 3), the MLP head architecture and training configuration (Section 3.4, Appendix Table 4), and the rank-percentile ensemble integration (Section 3.4), are all specified in sufficient detail to be reimplemented from the manuscript alone, using only the cited public resources and the official competition data.

Two derived artefacts are not released with this paper: the full unlabelled soundscape embedding cache (127,896 windows) and the trained MLP head weights. Both are reserved for a planned follow-up study and may be released in conjunction with that work, consistent with the competition's CC BY-NC-SA data-licence terms. Their construction is documented here in sufficient detail to reproduce without the released files; no result in this paper depends on private data or code beyond what is described and what is publicly cited.

4. Experiments and Results

This section reports the seven scored submissions that constitute the empirical core of the study, ordered chronologically. All scores are public-leaderboard macro-averaged ROC-AUC unless explicitly identified as private. Approximate ranks reported for individual submissions are snapshots taken at submission time against a field that grew to 4,092 teams by the final deadline; the final public and private ranks at the deadline are reported in Section 4.5.

4.1. From a Scratch Model to the Public Ceiling

My first submission was an EfficientNet-B0 classifier trained from scratch on log-mel spectrograms of the focal training recordings, with a multi-label objective and standard audio augmentations (a short single-fold training run completed in the time available). Despite reasonable behaviour during training, it scored 0.787 on the public leaderboard, an approximate rank of 2,803 of the field at the time. This submission is useful precisely as a baseline of what a late entrant achieves working alone with limited time: a functional but uncompetitive model. The corresponding rank places it well outside any medal zone.

Replacing this scratch model with a fork of a publicly shared pipeline produced a large and immediate gain. An initial fork of an earlier public variant scored 0.944 (Submission 2; rank $\approx$ 1,029 at submission time). A fork of the more thoroughly tuned exp019 variant scored 0.949 (Submission 3; rank $\approx$ 411 at submission time), placing it in the top $\approx$12% of the field. The aggregate movement from 0.787 to 0.949 represents an absolute ROC-AUC gain of $+0.162$, achieved without any model training of my

own between submissions one and three, and at negligible compute cost. This number quantifies the value of public-resource leverage for a late entrant in this competition.

At this point a structural feature of the leaderboard became noticeable and consequential. The score 0.949 is not a distinctive position but a dense plateau: at the final submission deadline, hundreds of teams shared the identical rounded public score of 0.949, with my submission's final public rank settling at 1,432 of 4,092 teams. Within the tied cluster, relative ranking is determined by submission timestamp rather than model quality (Section 7 of the official competition rules). Reaching 0.949 therefore places an entrant inside a large equivalence class of teams that have all converged on the same public pipeline; escaping it upward on the public leaderboard requires actually exceeding the shared ceiling, not merely matching it. This observation motivated the central experiments of Section 4.2.

4.2. The Custom-Head Ablation

I tested the most natural hypothesis available to a late entrant in this regime: that a small classifier head with a different inductive bias, trained on the same Perch v2 embeddings that the public pipeline already consumes, could provide useful ensemble diversity. The head was trained as described in Section 3.4 and produced, in isolation, behaviour that I found encouraging. Its best validation macro-ROC-AUC reached 0.852 (epoch 17) over the 37 scoreable classes in the held-out fold, with per-taxon median AUCs of 0.943 (Aves, 13 classes), 0.881 (Amphibia, 13 classes), 0.875 (Insecta, 9 classes), 0.928 (Reptilia, 1 class), and 0.983 (Mammalia, 1 class). The head's per-window predictions had a median per-class Spearman correlation of 0.581 with the raw Perch logits on a 2,000-window sample of the unlabeled training soundscapes. This is fair, but well short of the perfect correlation that would indicate that the head had simply learned to echo its inputs.

Notwithstanding these favourable standalone signals, the head failed to improve the ensemble in any configuration I tested. At a uniform blend weight of 0.07, the public score fell from 0.949 to 0.947 (Submission 4). Lowering the weight to 0.03 recovered part of the loss, to 0.948 (Submission 5). Restricting the head's contribution to the 162 Aves classes (which happens to be its strongest taxon by both validation AUC and Perch's underlying representational quality) also yielded 0.948 (Submission 6). The pattern is monotone and consistent: every positive amount of head contribution degraded the ensemble, with degradation scaling with weight. When extrapolated to zero weight, the trend passes through the 0.949 baseline. No configuration matched, let alone exceeded, the baseline.

I interpret this result in Section 5. Summarily, the prototype sequence model already in the public pipeline consumes the same embeddings, and the head's apparent diversity reflects noise rather than a complementary signal that is label-relevant, despite being real in the output space.

4.3. A Pre-registered Blend-Ratio Probe

Because the head experiments suggested the embedding-derived view of the data was already saturated by ProtoSSM, I ran one further test motivated by the opposite hypothesis, which is that marginally shifting the primary blend toward the spectrogram-based SED component, which represents the most orthogonal view available within the public pipeline, might help somewhat. I changed the ProtoSSM/SED blend ratio from its tuned default of 0.60/0.40 to 0.55/0.45, with the head disabled to isolate the variable. Before submitting I pre-registered the decision rule, which is retain the change only if the new public score strictly exceeded 0.949; otherwise discard it. The submission scored exactly 0.949 (Submission 7), which was identical to the baseline at three-decimal places. Per the pre-registered rule I discarded the change as an improvement, while retaining it as my second final submission for the hedging reason given in Section 3.6.

4.4. Runtime Budget

All scored submissions ran within the 90-minute CPU runtime budget. The baseline pipeline (Submission 3) ran in approximately 41 minutes on the scoring infrastructure. The head-augmented submissions added between 2 and 10 minutes of CPU runtime (Submissions 4 - 6 ran in 43, 51, and 43 minutes

respectively), driven mostly by the per-class rank-transform of the head's outputs rather than by head inference itself. I highlight this because it demonstrates the head integration is not constrained by the competition's compute budget: ample headroom remains for further ensemble components, were such components available.

4.5. Final Public and Private Leaderboard Results

For traceability, Table 2 records the provenance of the official results: the competing identity, the two selected final submissions, and the public and private leaderboard scores and ranks as recorded on the official Kaggle competition leaderboard. All figures are drawn from the official post-deadline leaderboard for BirdCLEF+ 2026 [1].

Item	Value
Competition	BirdCLEF+ 2026 (Kaggle code competition)
Team / participant handle	`didianwanane`
Final submissions selected	Submission 3 (exp019 fork, 0.60/0.40 blend); Submission 7 (blend ratio 0.55/0.45, no head)
Public LB score (both finals)	0.949
Final public LB rank	1,432 of 4,092
Official (private) score	0.94063
Final private LB rank	1,313 of 4,092 (top 32.1%)
Public-to-private movement	+119 places

Table 2: Provenance of the reported official results. Scores are macro-averaged ROC-AUC on the public and private test splits, as recorded on the official Kaggle BirdCLEF+ 2026 leaderboard. The two final submissions were tied at the public score; the better private score determines the official standing.

The competition closed with 4,092 teams. Both my finals reached the public score of 0.949, leaving me at public rank 1,432: within so dense a tie, ordering is by submission timestamp, and many teams reached the same rounded score earlier. The private leaderboard then resolved the tie, placing me at 0.94063 and rank 1,313, a climb of 119 places. The competition reports only the better of an entrant's two finals, so I cannot verify which produced the 0.94063 figure; what is verifiable is the direction, which is that within a cluster whose members moved both ways on private, my submission moved up.

I stress the modesty of this. A 119-place gain is a 2.9% rank-percentile improvement, and 0.94063 sits well below the medal threshold (top 10%, roughly the top 409 teams). I report it not as an achievement but as a data point for the paper's argument about defensive submission selection: the cluster I described as a single rounded public number did disperse on private, and declining to over-tune to public signal produced a small but measurable benefit. The interpretation is developed in Section 5.2.

5. Discussion

5.1. Why the Head Did Not Help: Signal Redundancy in Foundation-Model Pipelines

The aspect of the head's failure which is most prominent is the dissonance between its standalone behaviour and its ensemble behaviour. By every measurement I could perform before integration, the head looked like a reasonable ensemble candidate - macro-ROC-AUC of 0.852 on real-labelled validation windows, per-taxon AUC medians above 0.87 in every taxon with more than one scoreable class, and a median Spearman correlation of only 0.58 with its input Perch logits. Classical decompositions of ensemble error treat such moderate output decorrelation as the precondition for ensemble lift [9, 8]: an ensemble of classifiers improves over its average member to the extent that the members' errors are uncorrelated. By this conventional reading, the head should have helped. However, it did not.

I propose that the conventional reading conflates two distinct quantities: diversity in outputs and diversity in errors. The two coincide in the classical ensemble setting where each member sees the raw

input directly and applies its own inductive bias. They can come apart in foundation-model pipelines where multiple downstream components consume the same upstream representation. In such pipelines the foundation model has already filtered the raw audio into a particular feature space, retaining the information it judges relevant and discarding the rest. Downstream classifiers on that representation are not exploring the input space independently; they are all reasoning over the same filtered view. Their outputs can decorrelate freely (different architectures, different training data, different regularisation), but their access to label-relevant information has already been bottlenecked upstream.

In this account, the head's measured 0.58 Spearman correlation with the raw Perch logits reflects two combined effects: a genuine difference in how the head processes the embedding space (a function of architecture, training data, and regularisation), and noise around the head's underlying signal. Importantly, only the first of these effects can usefully complement another classifier already operating on the same embeddings. The second, which is noise, is exactly what an ensemble blend will amplify rather than cancel, because it is independent of the head's training objective but not independent of the prediction it produces.

The monotone weight-sweep pattern observed in Section 4.2 is consistent with this account, and difficult to reconcile with the alternative hypothesis that the head adds useful signal that is merely outweighed by ProtoSSM's accuracy. Under the useful-signal hypothesis, the optimal blend weight would be small but strictly positive, and the LB curve as a function of head weight would have a local maximum away from zero. Conversely, what I observed is monotone degradation through the tested range, extrapolating cleanly through the baseline at zero weight. This is the trademark of a component that contributes pure noise to the blend, not small signal in a large noise envelope. The class-specific variant (restricted to Aves, the head's strongest and best-validated taxon) showed the same pattern, ruling out the possibility that the failure reflects head weakness on non-bird taxa alone.

This interpretation has two practical consequences. First, standalone validation performance is an unreliable predictor of ensemble lift in this setting: the head's 0.852 validation AUC was a real measurement of its competence at the task, but it told me nothing about how its errors related to those of the model already in the ensemble. Second, the path likely to add value involves representational orthogonality rather than architectural variety on a shared representation. The baseline pipeline itself exemplifies this: ProtoSSM (operating on Perch embeddings) and the distilled SED ensemble (operating on mel-spectrograms) blend productively, because their access to label-relevant information truly differs. A model on raw waveforms, a model on a different foundation embedding (e.g., from a spectrogram-trained backbone), or a model that consumes long-context audio rather than 5-second windows would all be plausible candidates for the kind of orthogonality the head lacked. I did not have time to train any of these within the competition window, but they are the concrete next steps suggested by the analysis.

This outcome is also consistent with prior evidence on how much downstream architecture matters once a strong bioacoustic embedding is fixed. Ghani et al. [10] show that simple linear probes on frozen Perch or BirdNET embeddings already achieve high transfer performance across diverse bioacoustic tasks, with the choice of embedding mattering far more than the choice of downstream classifier. Read through the lens of the present result, this is the expected pattern: when the embedding has already concentrated the label-relevant information, a more expressive head (my two-layer MLP) recovers little beyond what a linear probe extracts, and therefore has little complementary signal to contribute to an ensemble already built on the same embedding. A natural question, raised in review, is whether a head with a markedly different decision geometry, for example a k-nearest-neighbour classifier in the embedding space rather than a parametric MLP, would produce errors structured differently enough to complement the existing pipeline. I did not test this, and it is a worthwhile direction; but I would expect the same upstream bottleneck to bound its contribution, since it too consumes only the frozen Perch representation.

5.2. The Public-Score Cluster and the Shake-Up: An Empirical Test

A second contribution of this study is empirical: the documentation of the dense public-score cluster at 0.949 and the direct observation of its dispersal on the private leaderboard. Where Section 5.1 explains

why the most natural late-entry intervention (a custom embedding head) fails to add ensemble value, the present section turns to the structural question of what an entrant should do, given that knowledge, when locked inside such a cluster.

For competitors, the immediate implication is that the marginal value of public-leaderboard tuning, once inside the tied cluster, is bounded by the noise of the public-private split. With several hundred teams clustered at the same rounded score, the score differences at the fourth decimal place approach the resolution noise of the public test subset itself. Configurations that look incrementally better on the public sample may be fitting that sample at the expense of private-leaderboard robustness. This is an instance of the leaderboard-overfitting phenomenon studied empirically by Roelofs et al. [11] across over a hundred Kaggle competitions. I made the explicit choice to stop tuning at the baseline rather than risk this trade-off. My two final submissions (the tuned baseline and one principled small variant) were intended to provide independent draws from the private-test distribution while minimising overfitting risk.

The final-leaderboard outcome (Section 4.5) provides direct empirical support for this strategy. The public 0.949 cluster did indeed disperse on the private leaderboard: the same cluster members ended at different private scores, with my own submission landing at 0.94063 and rank 1,313 of 4,092. Relative to my final public position (rank 1,432), the shake-up produced a net climb of 119 places. The single-team observation is not, on its own, decisive evidence for the general claim that defensive submission selection benefits late entrants in expectation. That is to say, with a sample of one entrant I cannot rule out the possibility that the climb reflects luck rather than the predicted mechanism. What the observation does establish is that the predicted mechanism was operative. Specifically, the cluster dispersed, with non-trivial inter-team movement, and that for at least this one entrant the disciplined non-tuning choice did not produce a private-leaderboard penalty.

For the competition's broader interpretation, the cluster size indexes how thoroughly the public-resource recipe has been exhausted. When hundreds of teams converge on the same public score using the same pipeline, the public leaderboard no longer ranks them by model quality but by submission timestamp, and distinguishing genuine model differences then requires either higher-resolution scoring or an expectation of significant public-to-private shake-up of the kind observed here.

5.3. Recommendations for Late Entrants

Piecing these two aspects together, I offer concrete recommendations to late entrants in tightly-clustered, foundation-model-driven competitions:

- **Fork first, model second.** With less than several weeks remaining, reaching the public ceiling via a tuned public pipeline dominates any same-budget training effort; the gain I measured was $+0.162$ absolute ROC-AUC.
- **Diagnose the cluster.** If hundreds of teams sit at the identical public score, that score is the recipe's ceiling, not a starting point for tuning.
- **Distrust standalone validation as a predictor of ensemble lift.** What matters is not a candidate component's intrinsic quality but the structure of its errors relative to what the ensemble already contains.
- **Prefer representational orthogonality.** Add a component that consumes a different representation (mel-spectrograms, raw waveforms, alternative embeddings); a second classifier on the same foundation-model embeddings is, on this evidence, the wrong place to look.
- **Use the two-submission hedge thoughtfully.** Two distinct prediction files that both reach the public ceiling give independent draws from the private-test distribution at no public-score cost.
- **Stop when the evidence says stop.** Tuning that does not exceed the baseline at three-decimal resolution risks adverse shake-up rather than gain.

5.4. Limitations

The central limitation of this study is that my negative result rests on a single head architecture, a single pseudo-labelling strategy, and a single integration scheme (rank-percentile blending). Stronger evidence would require comparison against

- heads with substantially different inductive biases (e.g., a small transformer over the per-file sequence of embedding windows),
- alternative pseudo-label strategies (e.g., distilled targets from the full public pipeline rather than raw Perch logits), and
- alternative integration schemes (e.g., stacking via a learned meta-model rather than rank-percentile blending). Time and compute did not permit these extensions within the competition window. My finding is therefore best read as a clean, well-documented negative result for the most natural version of the hypothesis, not as a definitive refutation of all forms of foundation-model-head ensemble augmentation.

A second limitation concerns the strength of the empirical claim in Section 5.2. With a single entrant, the observed 119-place private-leaderboard climb relative to the final public position is consistent with the predicted mechanism (cluster dispersal) but does not on its own demonstrate that defensive submission selection is generally beneficial for late entrants. A stronger empirical test would require pooled data across many entrants in similarly clustered competitions, ideally with information about each team's degree of public-leaderboard tuning. I report the single-team observation transparently as a data point rather than as definitive support.

A third limitation is the small size of the real-labelled validation set (144 windows spanning 37 of the 234 classes) which constrained the granularity of my standalone evaluation of the head. The competition's strong asymmetry between abundant unlabelled soundscapes and scarce expertly-labelled ones is a structural property of passive-acoustic monitoring data and remains a methodological challenge across the BirdCLEF series.

6. Conclusion

This paper documented a late-entry approach to BirdCLEF+ 2026 built around two questions: how far public-resource leverage takes a late entrant, and whether a custom head on the same foundation-model embeddings can add value beyond the public pipeline. The answers are sharp. Leverage delivers an absolute ROC-AUC gain of $+0.162$ at negligible compute cost, while a custom MLP head on the same Perch v2 embeddings fails to improve the ensemble at any of three blend configurations, and a pre-registered probe of the blend ratio is exactly neutral. I read the negative result as signal redundancy: a downstream classifier on a representation that another pipeline component already consumes tends to add noise rather than complementary signal, however well it performs in isolation. The dense public-score cluster and its dispersal on the private leaderboard (Section 4.5) reinforce the practical lesson: once a public recipe is mature enough that hundreds of teams converge on one score, further public-leaderboard tuning has non-positive expected value, and the route upward is representational orthogonality rather than incremental tweaking.

The same analysis points to its own next step. The in-domain embedding cache built here (Section 3.3) is not itself a result, but it is the resource the analysis identifies as missing: the starting point for pairing Perch embeddings with a model on a very different representation, such as raw waveforms or a complementary foundation embedding. Whether that orthogonality succeeds where a second embedding-space head did not is the question the next stage of this work is positioned to address.

Acknowledgments

I gratefully acknowledge the BirdCLEF+ 2026 organisers and the Bezos Earth Fund AI for Climate and Nature Grand Challenge for releasing the competition data and supporting the broader effort. I also

thank the Perch team [5] for releasing the underlying Perch v2 foundation model on which most of the pipeline I used depends.

This entry builds directly on the work of several open-source contributors on the Kaggle platform whose publicly shared notebooks and datasets, released under permissive licences during the competition's discussion-forum culture, made the baseline pipeline of this paper possible. I name each below, in the order of their direct contribution to the final submission:

- **Derek** (Kaggle handle @sunderekkiz) published the notebook "birdclef 2026 exp019 eos4 rank power 06",[1] which is the direct source of my final baseline submission. The full pipeline architecture described in Section 3.2 is Derek's work: the Perch-driven ProtoSSM, the multi-stage rank-percentile blend with sound-event detection, the Bayesian site-by-hour priors, the isotonic calibration, the five-gate post-processing chain, and the tuned default blend weights. It represents many cycles of iteration documented across a numbered sequence of incrementally refined public notebooks. My contribution sits as a thin attempted extension on top of this foundation.
- **Imaad Mahmood** (Kaggle handle @imaadmahmood) published the earlier notebook "BirdCLEF+ 2026 - Perch v2 + ProtoSSM · 0.925",[2] which I used as the basis for my first public-pipeline fork (Submission 2, public ROC-AUC 0.944) and which represents an earlier checkpoint in the same lineage of work that Derek subsequently refined.
- **Tucker Arrants** (Kaggle handle @tuckerarrants) published three datasets without which the pipeline could not run within the 90-minute CPU budget: a CPU-optimised Perch v2 ONNX export with the discrete Fourier transform removed,[3] the distilled sound-event-detection student ensemble used in the blend,[4] and a precomputed waveform cache of the training data.[5]
- **g john rao** (Kaggle handle @jaejohn) published the perch-meta dataset,[6] which provides cached Perch v2 embeddings and mapped logits for the 59 fully-labelled training soundscapes (708 windows). This was both an essential input to the baseline pipeline's in-notebook training and the existence-proof that motivated my own extension to the unlabelled soundscapes (Section 3.3).
- **Rishikesh Jani** (Kaggle handle @rishikeshjani) published an additional Perch v2 ONNX export adapted for BirdCLEF+ 2026,[7] used in the pipeline alongside the Tucker Arrants export.
- **yukiZ** (Kaggle handle @hideyukizushi) published the StratifiedGroupKFold split data ("SGKF|K-202604041716")[8] used for fold definitions throughout the pipeline, and the Bird26 reproduction training notebook ("Bird26|REPROD|Perch+Proto/ResidualSSM|Train|s7177")[9] whose outputs feed the ProtoSSM and ResidualSSM components.
- **NNMax** (Kaggle handle @ashok205) published the tf_wheels package,[10] which provides the TensorFlow wheels required for offline inference under the competition's no-internet runtime constraint.

The empirical observation that public-resource leverage delivers a $+0.162$ absolute ROC-AUC gain at negligible compute cost in Section 4.1 is, concretely, a measurement of the cumulative value of the work of these contributors. The paper's central negative result should be read as an attempt to extend, not displace, what these contributors have built.

[1] https://www.kaggle.com/code/sunderekkiz/birdclef-2026-exp019-eos4-rank-power-06
[2] https://www.kaggle.com/code/imaadmahmood/birdclef-2026-perch-v2-protossm-0-925
[3] https://www.kaggle.com/datasets/tuckerarrants/perch-v2-no-dft-onnx
[4] https://www.kaggle.com/datasets/tuckerarrants/bc2026-distilled-sed-public
[5] https://www.kaggle.com/datasets/tuckerarrants/birdclef-2026-waveform-cache
[6] https://www.kaggle.com/datasets/jaejohn/perch-meta
[7] https://www.kaggle.com/datasets/rishikeshjani/perch-onnx-for-birdclef-2026
[8] https://www.kaggle.com/datasets/hideyukizushi/sgkfk-202604041716
[9] https://www.kaggle.com/code/hideyukizushi/bird26-reprod-perch-proto-residualssm-train-s7177
[10] https://www.kaggle.com/code/ashok205/tf-wheels

Declaration on Generative AI

The author used an AI assistant (Anthropic Claude) for editing, LaTex formatting and literature review during the preparation of this paper. All experimental design, analysis, and conclusions are the author's own. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the accuracy, integrity, and originality of the work presented in this paper.

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, M. G. d. Reis, L. K. Sampaio Teles, K.-L. Schuchmann, J. L. M. M. Sugai, K. T. P. Azevedo, P. d. N. 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] 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. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF+ 2025: Multi-taxonomic sound identification in the middle magdalena, colombia, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 2909–2919. URL: https://ceur-ws.org/Vol-4038/.

[4] S. Kahl, T. Denton, H. Klinck, H. Reers, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF 2023: Automated bird species identification in eastern africa, in: Working Notes of CLEF 2023 – Conference and Labs of the Evaluation Forum, volume 3497 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3497/paper-164.pdf.

[5] B. van Merriënboer, V. Dumoulin, J. Hamer, L. Harrell, A. Burns, T. Denton, Perch 2.0: The bittern lesson for bioacoustics, arXiv preprint arXiv:2508.04665 (2025). URL: https://arxiv.org/abs/2508.04665.

[6] D. Stowell, Computational bioacoustics with deep learning: a review and roadmap, PeerJ 10 (2022) e13152. doi:10.7717/peerj.13152.

[7] S. Kahl, C. M. Wood, M. Eibl, H. Klinck, BirdNET: A deep learning solution for avian diversity monitoring, Ecological Informatics 61 (2021) 101236. doi:10.1016/j.ecoinf.2021.101236.

[8] T. G. Dietterich, Ensemble methods in machine learning, in: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems (MCS 2000), volume 1857 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2000, pp. 1–15. doi:10.1007/3-540-45014-9_1.

[9] A. Krogh, J. Vedelsby, Neural network ensembles, cross validation, and active learning, in: G. Tesauro, D. S. Touretzky, T. K. Leen (Eds.), Advances in Neural Information Processing Systems 7 (NeurIPS 1994), MIT Press, Cambridge, 1995, pp. 231–238.

[10] B. Ghani, T. Denton, S. Kahl, H. Klinck, Global birdsong embeddings enable superior transfer learning for bioacoustic classification, Scientific Reports 13 (2023) 22876. doi:10.1038/s41598-023-49989-z.

[11] R. Roelofs, V. Shankar, B. Recht, S. Fridovich-Keil, M. Hardt, J. Miller, L. Schmidt, A meta-analysis of overfitting in machine learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32 (NeurIPS 2019), 2019, pp. 9175–9185.

A. Detailed Hyperparameters and Pseudo-Label Configuration

A.1. Embedding Extraction

Perch v2 inference was performed using the publicly-available no-DFT ONNX export of the model, executed via the ONNX Runtime CPUExecutionProvider on Kaggle's free T4 GPU session (the ONNX export does not currently use the GPU during inference; I used a GPU session to ensure compute-resource consistency). Audio was loaded at 32 kHz mono, 60 seconds per file (padded with zeros if shorter), and split into twelve non-overlapping 5-second windows. Files were processed in batches of 16. Total wall-clock time for 10,658 files was 264.7 minutes (4.4 hours), with stable throughput of approximately 0.7 files per second across the entire run.

A.2. Pseudo-Label Configuration

Parameter	**Value**
Per-class threshold percentile	97th percentile of class's training logit distribution
Minimum threshold floor	1.5 (raw Perch logit)
Perch-blind class exclusion	classes with max Perch logit < 1.0 across all 127,896 windows receive no pseudo-labels (44 classes)
Top-K cap per window	5 (highest-scoring positives retained, others dropped)
Minimum positives per kept window	1
Sample weight, real-labelled	1.0
Sample weight, pseudo-labelled	0.3

Table 3: Pseudo-label generation parameters.

A.3. Head Training Hyperparameters

Parameter / Outcome	**Value**
Architecture	$1536 \to 512 \to 512 \to 234$ MLP, LayerNorm, GELU, Dropout 0.3
Total parameters	≈ 1.3 M
Optimiser	AdamW, lr 1e-3, weight decay 1e-4
LR schedule	Cosine annealing over 25 epochs, $\eta_{\min}$ = 1e-6
Batch size	1024
Gradient clipping	Norm 5.0
Validation split	GroupKFold ($k = 5$) by filename; fold 0 reported
Validation rows (real labels only)	144
Validation scoreable classes	37 of 234
Best validation epoch	17
Best validation macro-ROC-AUC	0.8524
Median per-class AUC (Aves)	0.943
Median per-class AUC (Amphibia)	0.881
Median per-class AUC (Insecta)	0.875
Median per-class AUC (Reptilia)	0.928 (single class)
Median per-class AUC (Mammalia)	0.983 (single class)
Median Spearman corr (head vs Perch logits)	0.581

Table 4: Head training hyperparameters and outcomes.

A.4. Submission Inference Runtimes

Submission	Wall time (min)
Submission 3: exp019 baseline	≈ 41
Submission 4: head, uniform $w = 0.07$	≈ 48
Submission 5: head, uniform $w = 0.03$	≈ 51
Submission 6: head, Aves-only $w = 0.07$	≈ 43
Submission 7: blend ratio 0.55/0.45 (no head)	≈ 41

Table 5: Observed wall-clock runtimes for scored submissions on the BirdCLEF+ 2026 CPU-only scoring infrastructure (90-minute budget).