# Listening to the Pantanal: Runtime-Constrained Distillation and Diversity-Aware Ensembling for BirdCLEF+ 2026

Rashmi Banthia[1,†], Stefano Morelli[2,†] and Aman Kapoor[3,†]

[1]*Independent Researcher, USA*
[2]*DGS, Italia*
[3]*Verticurl, India*

**Abstract**

BirdCLEF+ 2026 required participants to detect 234 bird, amphibian, mammal, reptile, and insect classes in 5-second windows from passive acoustic monitoring soundscapes recorded in the Brazilian Pantanal. The task combined severe focal-to-soundscape domain shift, scarce labeled target-domain data, and a 90-minute CPU-only Kaggle inference limit. We present a runtime-constrained ensemble that combines OpenVINO-deployed log-mel sound event detection (SED) models with a complementary Perch/ProtoSSM branch. The SED system uses Perch embedding distillation on labeled recordings and unlabeled soundscapes, CNN-based SED teacher-logit distillation, and sparse positive-only pseudo labels. The final system combined branch outputs with per-class rank fusion, avoiding learned calibration on the small labeled soundscape set. It achieved a private leaderboard score of 0.95313 and a public score of 0.95347, outperforming every individual branch on both splits. Ablations showed that conservative pseudo supervision and representation-level transfer were more reliable than aggressive pseudo labeling or direct Perch-logit distillation. More broadly, leaderboard transfer depended less on small local out-of-fold (OOF) gains than on preserving branch diversity, managing domain shift, and satisfying the deployment constraint. Our code is publicly available at https://github.com/rashmibanthia/birdclef_2026.

**Keywords**

BirdCLEF+ 2026, sound event detection, passive acoustic monitoring, domain adaptation, knowledge distillation, pseudo labeling, ensemble learning

## 1. Introduction

Monitoring biodiversity at scale is essential for understanding how ecosystems respond to climate change, habitat loss, and other forms of human disturbance. The Brazilian Pantanal - the world's largest tropical wetland - is an especially important setting for this problem [1]. It supports exceptional biodiversity, including more than 650 bird species, while recurring wildfires, agricultural expansion, and other development pressures have created an urgent need for scalable conservation tools [2, 3, 1]. Passive acoustic monitoring (PAM) complements traditional field surveys by using autonomous recording units to collect continuous environmental audio with minimal disturbance [4].

The BirdCLEF+ 2026 challenge focuses on automated species identification in Pantanal PAM recordings. We use the official "BirdCLEF+" name to reflect the task's expansion beyond birds to other vocal taxa: participants must predict the presence of 234 target classes, spanning birds, amphibians, mammals, reptiles, and insects, for 5-second segments within hidden test soundscapes as part of the LifeCLEF 2026 evaluation campaign [5, 6]. The hidden test set consists of approximately 600 one-minute Pantanal soundscapes. Participants must predict class probabilities for every 5-second segment and target taxon. Unlike much of the supervised training data, these recordings consist of long-duration passive acoustic monitoring (PAM) soundscapes that capture the entire surrounding environment without human

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

[†]These authors contributed equally.

rjain29@gmail.com (R. Banthia); ste.morelli@gmail.com (S. Morelli); a.kapoor1391@gmail.com (A. Kapoor)
https://www.linkedin.com/in/rashmibanthia/ (R. Banthia); https://www.linkedin.com/in/steubk/ (S. Morelli); https://www.linkedin.com/in/aman-kapoor-08294bb9/ (A. Kapoor)
0009-0003-7735-2696 (R. Banthia); 0009-0005-5135-0507 (S. Morelli); 0009-0005-9956-6796 (A. Kapoor)

intervention, rather than isolated focal recordings of target species.

This setting creates a difficult domain adaptation problem. The largest source of supervised data consists of short, directed recordings of individual species from sources such as Xeno-Canto and iNaturalist, while the evaluation data consists of noisy multi-species soundscapes recorded by passive sensors in the Pantanal [6]. The released training soundscape set has only 66 expert-labeled files out of 10,658, leaving most in-domain soundscapes available only as unlabeled audio. This creates a low-resource in-domain learning problem. Similar focal-to-PAM domain shifts have been identified as a central difficulty in recent BirdCLEF tasks [7, 8]. In contrast to focal recordings, PAM soundscapes contain overlapping species, environmental noise, and extended periods with weak or absent target vocalizations. This mismatch makes robust semi-supervised learning particularly challenging. The challenge therefore makes pseudo-label-based semi-supervised learning [9] and teacher distillation [10] attractive but potentially fragile.

The inference constraint is equally important. BirdCLEF+ 2026 submissions are executed as Kaggle notebooks under a 90-minute CPU-only runtime limit, with no GPU available for scoring [6]. This constraint rules out many straightforward large model ensembles and makes deployment engineering part of the modeling problem. We therefore treated computational efficiency as a first-class design constraint. Model architectures, deployment formats, and ensemble strategies were selected to satisfy the CPU-only runtime budget while preserving predictive diversity.

Our final system achieved a private leaderboard score of 0.95313 and a public leaderboard score of 0.95347. This working note describes a system built around controlled distillation, conservative pseudo-labeling, model diversity, and disciplined validation. We used Perch [11] both as a feature source and as a teacher signal, training SED models with Perch embedding distillation on labeled and unlabeled audio. We generated high-confidence pseudo labels for unlabeled soundscapes using heterogeneous teacher models, and used them with low weight as positive-only targets during student training. More broadly, we analyze a generalization gap between local out-of-fold validation and hidden leaderboard results: local validation was useful for detecting broken runs, but unreliable for fine-grained optimization, and several changes with strong local validation signals failed to transfer to the hidden test set.

Our contributions are fourfold. First, we analyze Perch-based distillation and unlabeled soundscape utilization under severe domain shift. Second, we report ablation evidence showing that pseudo-label pressure and teacher diversity matter more than pseudo-label quantity alone. Third, we study validation failure modes and ensemble diversity, including cases where higher local OOF scores did not improve leaderboard performance. Fourth, we describe a runtime-constrained multi-branch BirdCLEF+ 2026 system that integrates a Perch-based State Space Model (SSM) branch derived from ProtoSSM-style public work [12] with OpenVINO SED models [13].

## 2. Competition Setup and Data

### 2.1. Task and Evaluation

BirdCLEF+ 2026 asks participants to detect 234 species from passive soundscape recordings. The target classes span five taxonomic groups, with birds making up the clear majority (Table 1). For each hidden one-minute soundscape, participants submit class probabilities for every 5-second segment and target class. Submissions are evaluated with a macro-averaged ROC-AUC variant that skips classes with no true positive labels in the scoring set [6].

**Table 1. Distribution of the 234 target classes by taxonomic class.**

| Taxonomic class | Target classes |
|---|---|
| Aves | 162 |
| Amphibia | 35 |
| Insecta | 28 |
| Mammalia | 8 |
| Reptilia | 1 |

## 2.2. Dataset Composition

The provided data present a pronounced class and domain imbalance. The focal training metadata contains 35,549 short recordings from Xeno-Canto [14] and iNaturalist [15], while the target-domain soundscape directory contains 10,658 one-minute recordings. Only 66 of those soundscape files have expert segment labels; after deduplicating repeated CSV entries, these files yield 739 labeled 5-second windows in `train_soundscapes_labels.csv` [6]. Thus the training data contain abundant target-domain audio, but very little target-domain ground truth, motivating semi-supervised learning, distillation, and cautious validation. Table 2 summarizes the main data sources, label regimes, and roles they play in our system.

**Table 2. Summary of the main data sources and label availability.**

| Source or subset | Size | Label type | Role in this work |
|---|---|---|---|
| Focal training metadata | 35,549 short recordings | Species-level focal labels | Supervised source-domain training data |
| Target-domain soundscapes | 10,658 one-minute recordings | Mostly unlabeled | Target-domain audio for Perch embedding distillation and pseudo-labeling |
| Expert-labeled soundscape set | 66 files / 739 five-second windows | Segment labels | Scarce in-domain supervision for SED training |
| Fully labeled ProtoSSM subset (subset of 66 files) | 59 files / 708 windows | Complete 12-window labels | ProtoSSM training subset after excluding partially labeled files |
| Hidden test soundscapes | Approximately 600 one-minute files | Hidden labels | Public/private leaderboard evaluation |

Figure 1 illustrates representative 5-second log-mel spectrogram windows from focal training recordings across four target taxa. These examples show the structured vocal patterns available in the source-domain recordings before the harder shift to passive soundscapes.

The hidden test set adds further constraints. It consists of approximately 600 one-minute soundscape files split between public and private leaderboard scoring, and all submissions must run inside Kaggle notebooks under a 90-minute CPU-only limit [6]. The problem is therefore two things at once: a domain adaptation task, where models trained on clean focal recordings must transfer to noisy passive soundscapes, and a deployment task, where inference speed is as important as accuracy.

## 2.3. Validation Strategy

Given how scarce the labeled soundscape data was, we made a deliberate choice: maximize in-domain training signal, even at the cost of unbiased out-of-fold estimation.

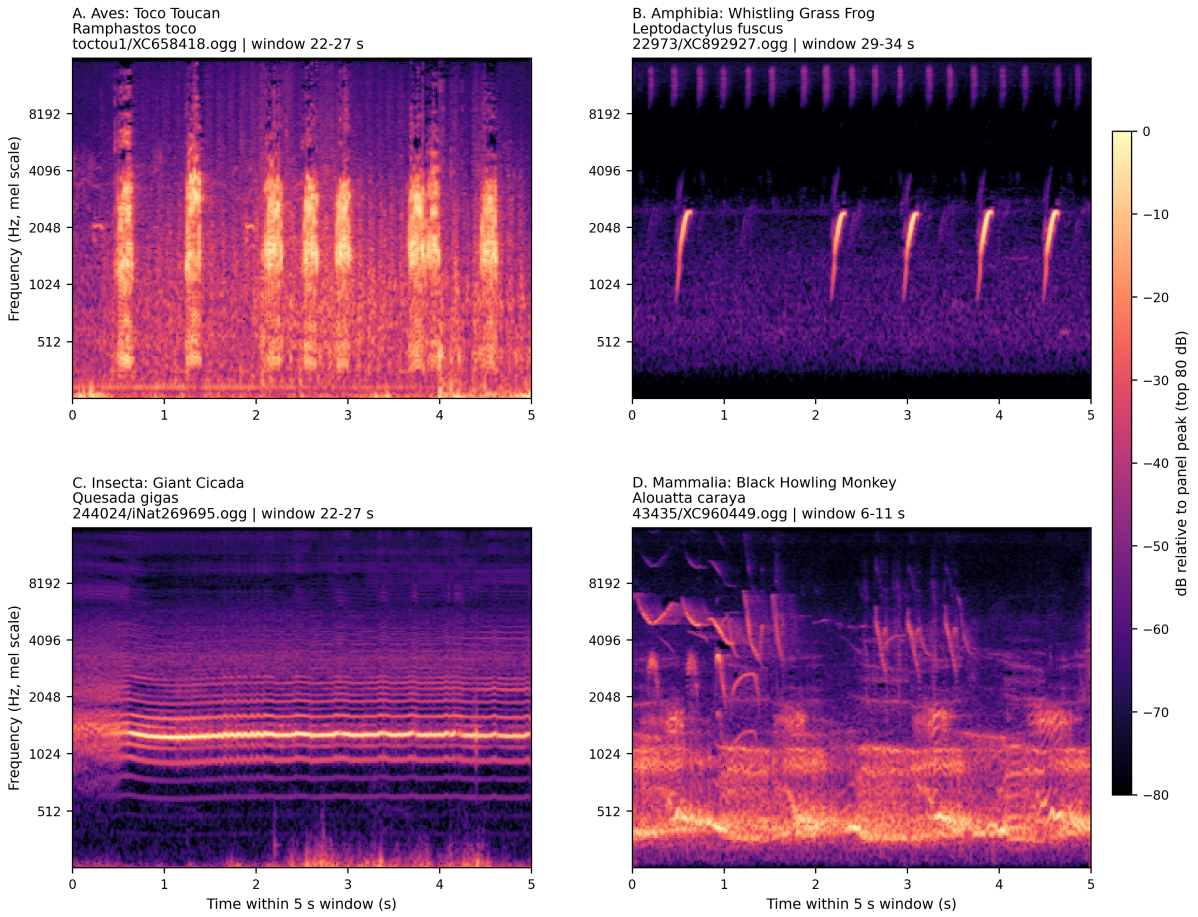


**Figure 1:** Example 5-second log-mel spectrogram windows from focal training recordings across four BirdCLEF+ target taxa. Each panel lists the taxonomic class, common and scientific name, source recording ID, and 5-second crop used for plotting.

For our Sound Event Detection (SED) models, which localize species calls within continuous audio, we selected checkpoints using three stratified folds built on focal audio. All 66 labeled soundscape files (739 deduplicated 5-second windows) were included in every fold's training set, never held out. This gave the models the best possible exposure to real soundscape conditions, but it also meant our validation scores reflected focal-domain performance only. In practice, these scores had low correlation with public leaderboard movement.

The ProtoSSM branch, adapted from a public Kaggle kernel [12], applied an even stricter filter: it trained only on the 59 soundscape files where all 12 windows carried labels (708 windows in total), and excluded the remaining seven partially labeled files. This kept the training labels clean at the cost of a slightly smaller in-domain set.

Across both branches, internal validation was most useful for detecting broken pipelines and misconfigured runs. Because it indicated little about the true performance of our model, we used it primarily as a pipeline sanity check rather than as evidence of hidden-test performance.

# 3. Methods

## 3.1. System Overview

The final inference ensemble was designed around two complementary branches, motivated by the observation that no single model family could jointly satisfy the domain transfer and runtime constraints imposed by the competition setting: the SED branch provided fast CNN-based detectors over 5-second

acoustic events, while the Perch/ProtoSSM branch contributed a different feature space and stronger pretrained bioacoustic priors.

The SED branch consisted of compact log-mel CNN models that we exported to OpenVINO for CPU-only inference [13]. Given the 90-minute runtime budget, we treated computational efficiency as a binding constraint that shaped every architectural decision. To enable the use of multiple SED models within this budget, we decoded each one-minute soundscape once, constructed a single shared 5-second log-mel batch, and reused it across all SED models. Under this scheme, adding an additional CNN stream increased only forward-pass cost, without repeating audio decoding or spectrogram construction: a non-trivial saving at inference scale.

For the second branch, we chose to contribute a distinct feature space rather than an additional log-mel view of the same audio. We applied Perch v2 [11] to the same 5-second windows, extracting dense bioacoustic embeddings and mapping its native taxonomy logits into the competition label space where possible. We fed these outputs into a ProtoSSM-style pipeline adapted from a public Kaggle kernel [12], with per-class probes, site/hour priors, and residual correction. We retained this branch in the final ensemble even when its standalone performance was not the highest among individual systems. Its contribution operated primarily through error decorrelation: because its predictions were less correlated with those of the CNN-based models, it improved rank-fusion ensemble performance beyond what individual accuracy alone would predict. In settings where labeled in-domain data is scarce, we found that this kind of predictive diversity proved more reliable than optimizing a single model family.

Table 3 summarizes the model families used in the final inference ensemble. We use model IDs (`M1`–`M5`) together with descriptive branch names.

**Table 3. Final ensemble branches.**

| ID | Branch | Model family | Ckpts. | Main teacher or feature source | PL targets | Weight |
|---|---|---|---|---|---|---|
| M1 | RegNetY-032 pseudo-label SED | RegNetY-032 | 3 | ConvNeXtV2-tiny teacher | Yes | 0.18 |
| M2 | HGNetV2-B4 SED | HGNetV2-B4 | 3 | ECA-NFNet-L0 teacher | No | 0.18 |
| M3 | EfficientNet-B0 SED | EfficientNet-B0 | 6 | ECA-NFNet-L0 teacher | No | 0.18 |
| M4 | HGNetV2-B4 diversity SED | HGNetV2-B4 | 1 | 0.6 SEResNeXt26t + 0.4 ConvNeXtV2-tiny teacher blend | No | 0.06 |
| M5 | ProtoSSM/Perch branch | Perch embeddings with SSM, MLP probes, and residual SSM | 1 branch | Perch v2 logits and embeddings | No | 0.40 |

Figure 2 summarizes the SED training signal flow. Hard labels supervised the classifier on focal recordings and labeled soundscapes, Perch embeddings shaped the representation on both labeled and unlabeled audio, CNN-based SED teacher logits supplied output-level distillation, and selected pseudo labels contributed only positive evidence.

## 3.2. SED Branch

### 3.2.1. Architectures

The SED branch consisted of several compact CNN streams. Each stream predicted clip-level and frame-level species probabilities for 5-second windows, and the final branch-level prediction was obtained by averaging checkpoints within the same stream. Model implementations and pretrained weights came from the `timm` model zoo where available, including the HGNetV2 variants [16]. The other named backbone families were standard computer-vision CNNs adapted to log-mel spectrogram inputs: EfficientNet [17], RegNet [18], ConvNeXt V2 [19], SEResNeXt through ResNeXt and squeeze-and-excitation components [20, 21], and ECA-NFNet through NFNet with efficient channel attention [22, 23].

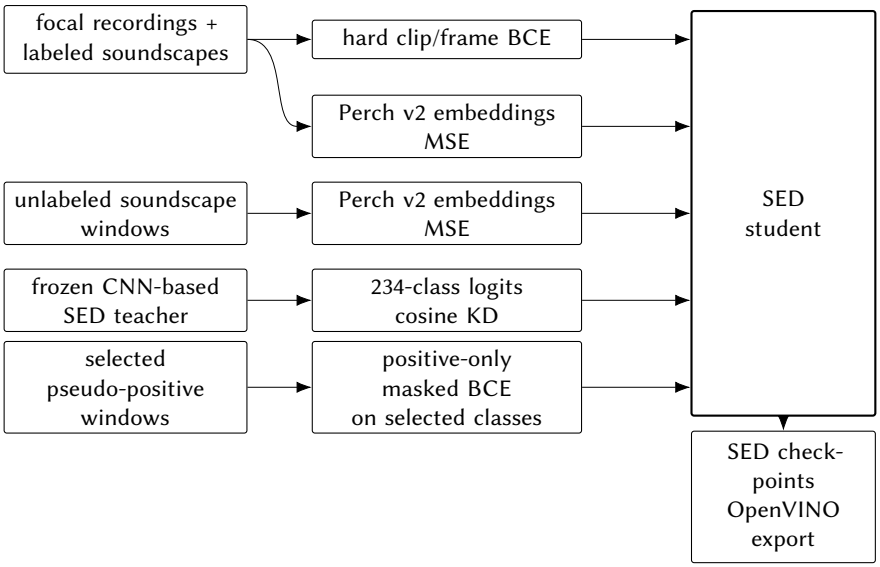


**Figure 2:** SED training signal flow. Hard labels, Perch embedding distillation, CNN-teacher logits, and selected pseudo positives provide separate training signals before OpenVINO export.

The final SED streams were:

- M1, RegNetY-032 pseudo-label SED: 3 checkpoints, distilled from a ConvNeXtV2-tiny teacher and trained with pseudo labels.
- M2, HGNetV2-B4 SED with ECA-NFNet teacher: 3 checkpoints, one per fold, distilled from an ECA-NFNet-L0 teacher.
- M3, EfficientNet-B0 SED: 6 checkpoints, corresponding to two seeds across three folds, distilled from an ECA-NFNet-L0 teacher.
- M4, HGNetV2-B4 diversity SED: 1 checkpoint, trained as a low-weight diversity stream with a blended SEResNeXt26t and ConvNeXtV2-tiny teacher.

The HGNetV2-B4 diversity training run also produced a standalone three-fold SED model. In the submitted ensemble, however, M4 used only fold 1 as a low-weight diversity stream.

#### 3.2.2. Perch Embedding Distillation

We initialized several SED branches from source-domain pretraining on prior bioacoustic datasets, including the official BirdCLEF 2021–2025 challenge data releases [24, 25, 26, 27, 28] and Nikita Babych's BirdCLEF 2025 extra species/target dataset [29]. Before source pretraining, we excluded classes overlapping the BirdCLEF+ 2026 target set [6]. We used these corpora exclusively as initialization: BirdCLEF+ 2026 labels, soundscape labels, Perch targets, and pseudo labels were introduced only during the task-specific training stage.

In both teacher and student SED training, Perch v2 [11] provided a 1536-dimensional embedding for each 5-second crop. We trained the SED model to project its internal features into the same embedding space, optimizing a mean-squared-error (MSE) loss against the Perch embeddings. We applied this distillation loss to both labeled audio and unlabeled soundscape windows, giving the student access to Perch's bioacoustic representation regardless of whether ground-truth labels were available.

Figure 3 visualizes these Perch embeddings for focal training recordings with an exploratory t-SNE projection [30]. The visualization illustrates structure in the pretrained embedding space; t-SNE distances were not used for model selection.

We used the Perch embedding MSE loss to transfer representation structure rather than class decisions. This distinction mattered because the competition target set and the Perch output label space did not align perfectly. By distilling embeddings rather than directly forcing agreement with Perch logits,

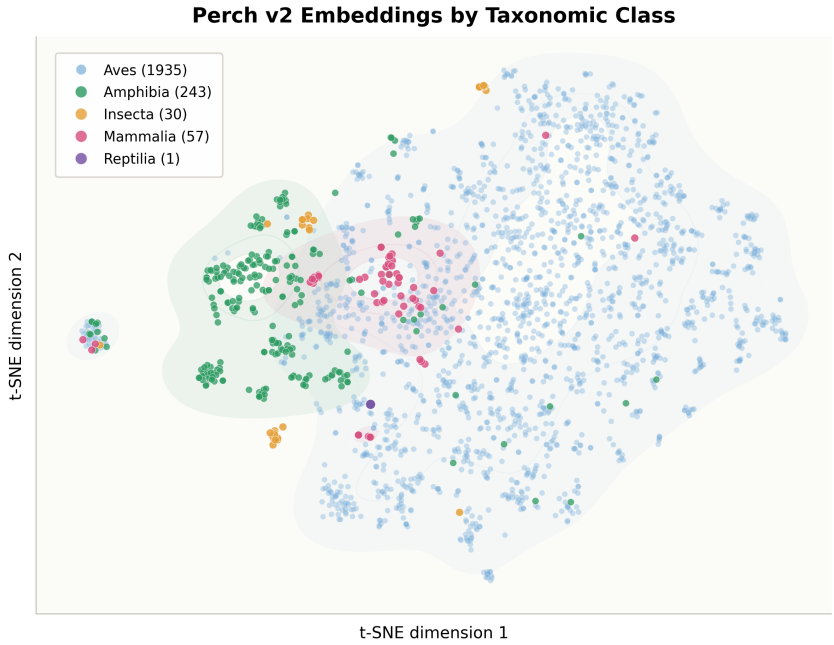


**Figure 3:** Exploratory t-SNE projection of Perch v2 embeddings from focal training recordings, colored by taxonomic class. Each point is one 5-second peak-energy window from a single recording, with up to 12 recordings sampled per species (2,266 points total); legend values are the resulting per-class point counts.

the SED models could benefit from Perch's pretrained bioacoustic representation while preserving task-specific classifier learning from BirdCLEF+ labels.

### 3.2.3. Teacher-Student Training

The SED supervision is most clearly described by separating teacher training from student training (Table 4).

**Table 4. Data sources and loss terms for SED teacher and student training.**

| Data source | Teacher SED training | Student SED training |
|---|---|---|
| Focal recordings | Hard-label BCE on clip and frame-max logits; Perch embedding MSE | Hard-label BCE on clip and frame-max logits; Perch embedding MSE; cosine KD from a frozen CNN-based SED teacher or blend |
| Labeled soundscapes | Hard-label BCE on clip and frame-max logits; Perch embedding MSE | Hard-label BCE on clip and frame-max logits; Perch embedding MSE |
| Unlabeled soundscapes | Perch embedding MSE only; no hard labels | Perch embedding MSE only, except for selected pseudo-label windows |
| Selected pseudo-label windows | Not used for the teacher branches described here | Positive-only masked BCE on selected classes, with low loss weight and no pseudo negatives |

For hard labels, the SED head produced both clip-level logits and frame-level logits; we computed

the supervised loss as the average of BCE on the clip prediction and BCE on the maximum frame-level prediction. For teacher–student knowledge distillation [10], each student SED branch used a CNN-based SED teacher model or teacher blend. The teacher model was trained first, frozen in evaluation mode, and applied to the same 5-second training crops used by the student. The student then matched the teacher's clip-level logits with a cosine loss on the same log-mel input. Ground-truth BCE remained the main source of label supervision, while the teacher loss served as a low-weight guide that encouraged the student to follow similar class-ranking patterns. We did not treat the teacher outputs as hard pseudo labels. We used cosine similarity rather than KL divergence because it is less sensitive to logit-scale differences across teacher and student architectures and emphasizes the direction of the class-response pattern rather than calibrated probability magnitudes. This separated two distinct transfer signals: Perch shaped the backbone features through representation-level distillation, while cosine knowledge distillation from the CNN-based SED teacher shaped the classifier outputs. Shared SED training settings for M1–M4 are summarized in Appendix Table A2.

For the final SED branches, the task-specific objective used fixed loss weights:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{hard}} + \mathcal{L}_{\text{perch,labeled}} + \mathcal{L}_{\text{perch,unlabeled}} \\ &\quad + 0.3\,\mathcal{L}_{\text{cnn-kd}} + \lambda_{\text{pl}}\mathcal{L}_{\text{pseudo}}, \\ \mathcal{L}_{\text{hard}} &= 0.5\ \text{BCE}(z_{\text{clip}}, y) + 0.5\ \text{BCE}(\max_t z_{\text{frame},t}, y), \\ \mathcal{L}_{\text{cnn-kd}} &= 1 - \text{cosine}(z_{\text{student}}, z_{\text{teacher}}).\end{aligned}$$

The two Perch terms are embedding-distillation losses on 1536-dimensional Perch targets. For non-pseudo-label branches, $\lambda_{\text{pl}} = 0$. For M1, $\lambda_{\text{pl}}$ ramped to 0.05 after epoch 6 over a 5-epoch ramp, and the pseudo side loader was sampled on 5% of main training steps after the ramp.

### 3.2.4. Pseudo-Label Training

We used pseudo labels [9] conservatively and only in selected SED branches. Rather than treating them as complete annotations, we treated pseudo labels as positive evidence only: for a given unlabeled soundscape window, only the selected classes contributed to the loss, and all unselected classes were masked rather than assigned as negatives.

In the final ensemble, this mechanism appeared exclusively in M1, the RegNetY-032 pseudo-label branch. M1 used the pseudo-label set through a separate side loader sampled on only 5% of training steps, with a pseudo-label loss weight of 0.05. This design gave the student additional exposure to target-domain positives while reducing the risk of suppressing real but unlabeled species in dense multi-species soundscapes.

This conservative design was important because PAM soundscapes are weakly and incompletely labeled by nature. A species not selected by the pseudo-label generator is not necessarily absent. Treating unselected classes as negatives would therefore risk teaching the model that hidden vocalizations are absent, especially in dense multi-species windows. Positive-only pseudo supervision reduced this failure mode.

The pseudo-label artifact was generated offline before training the RegNetY-032 consumer. Candidate class-window positives were proposed using a logit-mean ensemble of two HGNet generators: an HGNetV2-B6 model trained without class pseudo labels and an HGNetV2-B3 model. A SEResNeXt26t model served only as a veto branch and did not contribute to the target probability.

We retained only high-confidence positives and applied caps to avoid concentrating pseudo labels in a small number of files, windows, or species. The final artifact contained 9,147 pseudo-positive targets across 7,402 windows, 1,645 files, and 42 classes. No pseudo labels were selected from hard-labeled soundscape files. These caps were the only balancing mechanism used during pseudo-label generation; the downstream M1 consumer added no further class balancing beyond its low sampling rate, low loss weight, delayed activation, and positive-only masking. Exact selection thresholds and caps are listed in Appendix A.

## 3.3. Perch/ProtoSSM Branch

### 3.3.1. Perch Features

We ran the same 5-second windows through an ONNX-exported Perch v2 model [31, 11], which produced native taxonomy logits and a 1536-dimensional dense bioacoustic embedding for each window. This branch contributed to the final ensemble a pretrained bioacoustic representation and a temporal/contextual modeling path structurally distinct from the log-mel CNN streams.

The Perch branch was valuable not only as a standalone predictor but also as a diversity source. It was trained and represented differently from the CNN SED branches, so its errors were less redundant with the log-mel CNN streams. This made it useful in the final rank-fusion ensemble even when some SED branches were stronger as standalone submissions.

Perch did not emit a native 234-class BirdCLEF+ head. The inference notebook mapped its native taxonomy logits into the competition label space where possible: 203 classes used direct eBird-code or scientific-name matches, 3 additional Amphibia classes used genus-level proxies, and the remaining 28 classes had no Perch logit signal.

### 3.3.2. ProtoSSM Pipeline

We fed the Perch outputs into a ProtoSSM-style pipeline adapted from a public Kaggle kernel [12]. The branch combined:

- a state-space classifier,
- site/hour context,
- a site/hour log-odds prior,
- per-class MLP probes over the Perch embeddings, and
- a residual state-space refiner.

The branch applied its own calibration and smoothing internally before writing its probability table. This made the ProtoSSM output a completed branch-level prediction rather than an intermediate feature table for downstream calibration. Table 5 summarizes its inference-time components.

**Table 5. ProtoSSM inference-time components.**

| Component | Input | Role in the branch |
|---|---|---|
| ONNX Perch v2 feature extractor | 5-second waveform windows | Produced 1536-dimensional embeddings and native taxonomy logits. |
| Taxonomy mapping and genus proxies | Perch logits plus competition taxonomy | Mapped direct scientific-name/eBird-code matches and three Amphibia genus proxies into the target space. |
| Site/hour prior tables | Soundscape labels and filename metadata | Added global, site, hour, and site-hour log-odds priors. |
| LightProtoSSM | Perch embeddings/scores and site/hour IDs | Modeled the 12 windows with bidirectional SSM layers, class prototypes, and context embeddings. |
| Per-class MLP classifiers | Perch embeddings and mapped scores | Added learned corrections for classes with enough training signal; 58 classifiers were loaded. |
| Residual SSM | Perch embeddings plus first-pass scores | Predicted additive residual corrections. |
| Internal calibration and smoothing | ProtoSSM branch scores | Applied temperature, file-confidence, and rank-aware scaling, adaptive smoothing, and per-class thresholds. |

## 3.4. Inference and Runtime Optimization

### 3.4.1. Shared Audio Front-End

We decoded each one-minute soundscape once and split it into twelve non-overlapping 5-second windows. From these windows, we computed a single log-mel spectrogram batch and reused it across all SED models. This design eliminated repeated audio decoding and spectrogram computation, making it feasible to run several CNN predictors within the CPU budget. Batch sharing was safe because all SED streams operated under the same mel configuration, which we verified at model loading time.

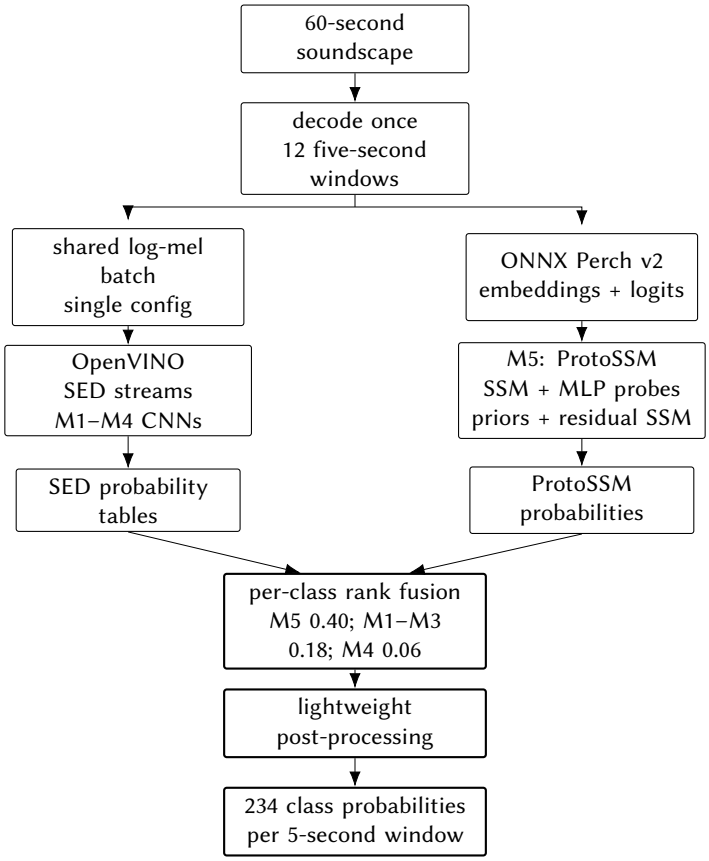


**Figure 4:** Final inference pipeline. A single decoded soundscape feeds the shared SED mel batch and the ONNX Perch/ProtoSSM branch; completed branch probability tables are then combined by per-class rank fusion and lightweight post-processing.

### 3.4.2. OpenVINO Deployment

Before submission inference, we exported the trained SED checkpoints to OpenVINO Intermediate Representation (IR), OpenVINO's deployable graph-and-weights model format, and compiled the resulting IR models for CPU execution inside the Kaggle notebook [13]. We passed the shared log-mel batch through the four SED streams, M1–M4, shown in Figure 4.

Within each stream, we averaged checkpoint outputs and applied temporal smoothing to the resulting probabilities across adjacent 5-second windows with weights 0.2 / 0.6 / 0.2. We then created one probability table per stream for the rank-level fusion described in Section 3.5, keeping each SED family separate until the final ensemble step: collapsing the CNN streams earlier would have reduced the diversity available to the rank fusion stage.

In total, the submitted ensemble ran 13 SED checkpoints across four CNN streams plus the Perch/ProtoSSM branch. It used nearly the full 90-minute CPU budget, making the shared audio front end, OpenVINO SED export, and ONNX Perch path necessary for the submission to finish.

## 3.5. Ensemble Design

### 3.5.1. Rank Fusion

We approached the final ensemble as a runtime and validation problem, not only as a model-averaging problem. The 90-minute CPU limit imposed two upstream decisions: we ran Perch once through the ONNX ProtoSSM path, and we computed one shared OpenVINO log-mel batch for the SED streams. Once those computationally expensive operations were fixed, the ensemble had to combine heterogeneous model families without adding a learned calibrator or tuning a large number of rules on the small labeled soundscape set.

We kept the branch boundary explicit by design. The ProtoSSM branch emitted a completed probability table after its own internal calibration and post-processing. Each SED stream likewise emitted a completed probability table after checkpoint averaging and local temporal smoothing. This boundary kept the final ensemble step focused on cross-branch fusion: rather than retuning ProtoSSM-specific rules at the merger stage, we combined already-formed predictions from each branch.

Figure 5 summarizes this final merger boundary.

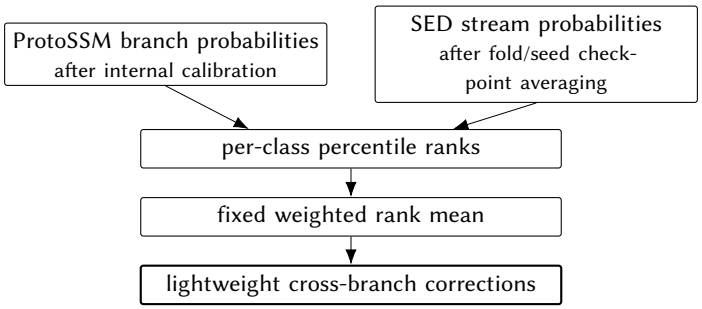


**Figure 5:** Rank-fusion merger. Completed ProtoSSM and SED probability tables are converted to per-class percentile ranks, combined with fixed weights, and passed through lightweight cross-branch corrections.

The final merger used per-class rank fusion rather than raw-probability averaging. For each target class, we converted every branch's predictions into percentile ranks over the submission rows, then averaged those ranks with the fixed branch weights shown in Table 3. This choice preserved the within-class ordering learned by each model while discarding absolute score scales - a deliberate tradeoff, given that Perch/ProtoSSM scores and CNN SED probabilities arose from different training objectives and the 66 labeled soundscape files were too few to support aggressive probability calibration. We set the weights by hand rather than by tuning on these labels: the public ProtoSSM kernel we adapted was a strong standalone system, which justified a large weight, but since our SED ensemble outperformed it we held ProtoSSM to a minority 0.40. The three-fold SED streams then shared equal weight (0.18), while the single-fold M4 stream received one-third of that (0.06), reflecting its smaller fold coverage.

### 3.5.2. Post-Processing

After rank fusion, we kept post-merge processing deliberately lightweight. We applied conditional rank corrections for cases where one model family produced an isolated high-confidence response without enough cross-family support, and we mirrored selected sonotype classes whose labels represented closely related acoustic variants. Because these corrections operated exclusively on completed branch outputs, they added negligible runtime and did not interfere with ProtoSSM's internal calibration path. We designed them to address recurring failure patterns while avoiding overfitting to the 66 labeled soundscape files.

The exact correction thresholds and sonotype groups are implementation details of the final merger and are listed in Appendix A.

# 4. Results

## 4.1. Final Leaderboard Results

Our final ensemble achieved a private leaderboard score of 0.95313 and a public score of 0.95347. More importantly, it outperformed every individual branch on both splits, validating the central design decision described in Section 3. We did not optimize for a single strongest standalone model. Instead, we combined branches whose errors differed, and the ensemble benefited from that disagreement.

**Table 6. Leaderboard scores for the final ensemble and major component branches.**

| Model or branch | Private score | Public score |
|---|---|---|
| Final ensemble | 0.95313 | 0.95347 |
| M1: RegNetY-032 pseudo-label SED | 0.94801 | 0.93405 |
| M4: HGNetV2-B4 diversity SED | 0.94608 | 0.93385 |
| M2: HGNetV2-B4 SED | 0.94572 | 0.93473 |
| M3: EfficientNet-B0 SED | 0.94238 | 0.93380 |
| M5: ProtoSSM/Perch branch | 0.92539 | 0.92874 |

Table 6 breaks down the standalone scores for each branch alongside the full ensemble. The gap between the best individual branch and the combined system was modest in absolute terms, but it appeared on both leaderboard splits. This consistency supports the interpretation that the gain came from complementary errors across branches rather than from tuning to the public split alone.

## 4.2. Training Ablations

We next examined which training signals produced leaderboard gains and which only improved local validation. The main pattern was that conservative, complementary signals transferred better than stronger versions of the same signal.

Table 7 summarizes the training-signal checks that most shaped the final recipe. Because all 66 labeled soundscape files were used during SED training, the OOF values are local three-fold focal-domain means rather than held-out soundscape estimates. We therefore used OOF mainly to compare related variants, and relied on standalone private/public leaderboard scores to judge transfer. The complete final ensemble roster is reported separately in Tables 3 and 6.

The checks fall into three recurring patterns. First, conservative positive-only pseudo labels helped when they were low-weight and branch-specific (M1), but stronger pseudo-label pressure did not reliably transfer. Second, teacher diversity was most reliable when the teacher roles were separated: Perch shaped the representation, while CNN teachers shaped output-level class rankings. In contrast, output-layer Perch KD and same-layer dual-teacher KD hurt transfer. Third, local OOF was useful for detecting viable runs, but several higher-OOF variants lost public leaderboard score, so final selection relied on leaderboard transfer and ensemble complementarity rather than OOF alone. Table 7 provides the supporting ablation checks.

**Table 7. Selected training-signal ablations and transfer checks.**

| Training signal check | Configuration | OOF | Priv. | Pub. |
|---|---|---|---|---|
| **Baseline** | RegNetY-032; CNN KD, no pseudo labels | 0.965 | 0.945 | 0.933 |
| **+ Positive-only pseudo labels** | M1: RegNetY-032; low-pressure positive-only pseudo labels | 0.966 | 0.948 | **0.934** |
| | Higher pseudo exposure: doubled pseudo-label sampling | 0.966 | 0.948 | 0.933 |
| **+ Kept SED branches** | M2: HGNetV2-B4; ECA-NFNet-L0 CNN-teacher KD; 3-checkpoint stream | 0.964 | 0.946 | **0.935** |
| | M3: EfficientNet-B0; ECA-NFNet-L0 CNN-teacher KD; 6-checkpoint stream | 0.959 | 0.942 | **0.934** |
| | M4: HGNetV2-B4; blended CNN-teacher KD | 0.966 | 0.946 | **0.934** |
| **OOF-only or not retained** | Sigmoid-BCE KD: alternate KD loss for M4 recipe | 0.968 | 0.948 | 0.933 |
| | CaFormer HGNet: CaFormer-S18 teacher [32] | 0.967 | 0.946 | 0.933 |
| **Negative transfer checks** | Perch logit KD: mapped Perch output KD | – | 0.943 | 0.928 |
| | Dual-teacher KD: mixed ECA-NFNet/HGNet-B6 output KD | – | 0.936 | 0.929 |
| | Full Perch MPL: dense Perch pseudo-label set | – | 0.927 | 0.910 |

Note: "–" indicates no comparable OOF under the same validation contract. Bold public scores mark branches retained in the final ensemble.

#### 4.2.1. Pseudo-Label Experiments

The pseudo-label experiments separated two very different uses of unlabeled target-domain audio. In M1, pseudo labels were used conservatively: only selected high-confidence positives were sampled through a low-frequency side loader, and unselected classes were masked rather than treated as negatives. This improved the RegNetY branch over the no-pseudo baseline on both private and public leaderboard splits. Increasing the pseudo-label sampling rate did not improve public transfer, suggesting that the useful signal came from sparse target-domain positives rather than from simply adding more pseudo-labeled examples.

The dense Perch pseudo-label diagnostic supported the same low-pressure interpretation. This diagnostic used a separate dense pseudo-label regime, not the sparse M1 pseudo-label artifact retained in the final ensemble. Here, MPL denotes Meta Pseudo Labels, where a teacher is updated using feedback from the student's labeled-data performance [33]. The full Perch MPL check used the full 127K Perch pseudo-label set with an approximately 5:1 pseudo-to-focal ratio, and its leaderboard score dropped sharply. In that setting, direct pseudo-label supervision may have dominated the real focal labels and amplified systematic errors in the pseudo-label source. A lighter curriculum variant was closer to neutral, and pseudo labels were more useful indirectly when they first improved a teacher model that then supplied KD to the student.

We attribute this pattern to label incompleteness. In multi-species PAM soundscapes, a pseudo-label generator will miss some real vocalizations, not necessarily because the species is absent, but because dense overlapping calls make exhaustive detection difficult. Treating unselected classes as negatives in that setting risks suppressing genuine but unlabeled species activity. Positive-only supervision reduces this failure mode by restricting the pseudo-label loss to confirmed detections, while low sampling frequency and low loss weight keep the signal subordinate to hard labels and Perch embedding distillation.

#### 4.2.2. Teacher Distillation Experiments

The teacher-distillation rows show that output-level KD was useful when the teacher role was well defined and separated from the Perch embedding objective. The four SED branches retained in the final ensemble all followed the same structural pattern: Perch embedding MSE shaped the backbone representation, while cosine KD from a CNN-based SED teacher shaped the classifier outputs. M3 and M2 paired this scheme with an ECA-NFNet-L0 teacher, M1 used a ConvNeXtV2-tiny teacher while adding low-pressure pseudo positives, and M4 used a blended teacher signal. In these branches, cosine KD copied the teacher's class-ranking pattern over the 234 target classes rather than calibrated probability magnitudes.

The ablations also revealed where this recipe broke down. M4 used a blended teacher signal and became valuable in the final ensemble, but the sigmoid-BCE KD variant achieved a higher local OOF while losing public leaderboard score relative to M4. The HGNet-B4 negative checks further showed that output-space Perch KD and same-layer dual-teacher KD were less reliable: Perch logits had incomplete target coverage, and multiplexing multiple CNN-based SED teachers may have created conflicting output supervision on ambiguous samples.

The overall pattern was consistent: teacher diversity was most reliable when teachers were separated by role or by representation level. Perch was most useful as a backbone embedding teacher. CNN-based SED teachers were most useful as output-level KD sources. When multiple teacher signals tried to shape the same output layer, the additional supervision did not reliably transfer.

### 4.3. Validation Failure Modes

The local validation protocol was useful for detecting broken runs, but it was not reliable enough for fine model selection. Several changes improved local OOF or looked locally safe, then failed to transfer to the hidden leaderboard. These failures shaped our final workflow: we used OOF to compare closely related runs and catch collapses, but we did not treat small OOF gains as sufficient evidence for production changes.

**Table 8. Representative local-to-leaderboard failures.**

| Diagnostic | Apparent signal | Leaderboard result | Likely failure mode |
|---|---|---|---|
| RegNetY source-pretrained teacher path | OOF 0.966592 (+0.000545 vs. M1) | Public 0.932, below related RegNetY references | OOF-only gain |
| Hard-labeled soundscape KD for SEResNeXt | Local gains up to +0.0044 macro and +0.0056 non-S22 | two runs: 0.917, 0.923 | labeled-soundscape overfit |
| 20-second HGNetV2 pseudo curriculum | Macro OOF flat/slightly positive; non-S22 positive | 0.913, below 0.929 HGNet anchors | pseudo-curriculum shift |
| Multi-head GeM pooling [34] | Pretrained backbone loaded, but neck widened $640 \rightarrow 1920$ | Private/Public 0.944 / 0.932, below M2 | head-compatibility mismatch |
| RMS-weighted focal crops | Small OOF change versus M1 (-0.000321) | 0.926, below M1 and baseline | crop-policy mismatch |

Note: OOF values are local validation scores, and leaderboard scores are standalone diagnostic submissions, not final-ensemble deltas. The non-S22 subset is defined in Appendix A. Rows summarize the signal that made each run plausible before the leaderboard check.

Across these diagnostics, the same pattern recurred. The RegNetY source-pretrained teacher path produced the best local OOF in its recent teacher line, but its public leaderboard score stayed below M1 and related RegNetY variants. The hard-labeled soundscape KD variants were especially misleading: they improved local macro and non-S22 validation, yet scored only 0.917 and 0.923 on the leaderboard. The 20-second pseudo-label curriculum and RMS-weighted focal crop sampler showed that locally neutral or near-neutral changes could still move the model away from the hidden-test distribution. The

multi-head GeM result was a structural warning: although pretrained weights loaded, concatenating GeM heads tripled the neck input width and left the wider neck undertrained.

These failures clarified the role of local validation rather than eliminating it. Local OOF helped us catch broken runs and compare narrow variants under the same validation contract, but the validation sets were too small and too distributionally specific to support aggressive recipe tuning. We therefore treated local validation as a sanity check, and promoted branches only when leaderboard transfer, runtime feasibility, and ensemble diversity also supported the change.

### 4.4. Ensemble Diversity Analysis

We assigned ProtoSSM a rank weight of 0.40 because it contributed complementary information to the ensemble. The fused system consistently outperformed every individual SED stream on the leaderboard, and our member-correlation audit suggested that ProtoSSM was less redundant with the log-mel CNN models than the CNN models were with one another. We treated this audit as approximate because ProtoSSM and the SED streams did not use identical fold and training contracts. However, the comparison was still informative for ensemble design: all SED streams shared the same fold strategy, so their mutual correlations were more directly comparable, while ProtoSSM's consistently lower correlation to every SED stream indicated a genuinely different prediction source rather than just another CNN variant.

To quantify this, we computed per-class Spearman correlations over 3,600 aligned prediction rows and 234 species from the Kaggle member-audit outputs. Correlations between ProtoSSM and the SED streams were about 0.410–0.429, whereas correlations among SED streams were much higher, about 0.698–0.817. ProtoSSM also had the lowest mean redundancy (0.420, compared with 0.648–0.691 for the SED members). Because of the fold-strategy mismatch, we did not use these values to tune weights directly; we used them as evidence for preserving a high-weight, structurally distinct ProtoSSM branch in the rank-fusion ensemble.

This also explains why we used rank fusion rather than a learned probability calibrator or a winner-take-all SED selection. Rank fusion preserved each branch's within-class ordering while reducing sensitivity to raw probability scale. That mattered because ProtoSSM probabilities and CNN SED probabilities came from different training objectives, and the labeled soundscape set was too small to calibrate those scales aggressively.

### 4.5. Lessons Learned

The experiments suggest five practical lessons for BirdCLEF+ style domain adaptation:

1. **Representation-level transfer was more reliable than output-level transfer.** Perch embedding distillation helped by shaping acoustic representations, while Perch logit distillation appeared more fragile because of target-space mismatch.
2. **Conservative pseudo labeling outperformed aggressive pseudo supervision.** Positive-only pseudo labels improved the RegNetY branch, while stronger pseudo pressure and dense soft targets were less reliable.
3. **Small OOF gains frequently failed to transfer.** Local validation was valuable for catching broken runs, but it was not trustworthy enough for fine-grained recipe selection under severe focal-to-PAM domain shift.
4. **Diversity mattered more than standalone branch strength.** The final ensemble gained from combining CNN SED streams with a Perch/ProtoSSM branch that was substantially less correlated with the SED cluster, instead of simply selecting the best individual SED model.
5. **Runtime constraints shaped successful modeling choices.** Shared preprocessing, OpenVINO deployment, and fixed rank fusion preserved branch diversity under the CPU-only scoring limit.

# 5. AI-Assisted Development

## 5.1. Experiment Management

We used LLM-based coding agents, including Codex [35] and Claude Code [36], for experiment tracking and operational coordination. The central artifact was a structured experiment log that grew to approximately 2,500 lines and covered many training, inference, ensemble, and deployment variants. The log connected internal experiment IDs to model families, training signals, artifacts, and outcomes: which runs used Perch embedding distillation, teacher-logit KD, positive-only pseudo labels, OpenVINO export, or a different validation contract; which Kaggle kernels or datasets belonged to each run; and whether the result was a local OOF improvement, a leaderboard improvement, a runtime check, or a rejected diagnostic.

This support was particularly valuable because the project evolved through many near-neighbor experiments whose internal names alone were not self-explanatory. For example, the assistants helped keep the relationship between baseline SED runs, pseudo-label consumers, diversity branches, and later merger variants explicit: which branch each run belonged to, what changed relative to its parent, what artifacts were produced, and whether a local OOF gain transferred to the leaderboard. They also helped convert informal experiment history into ablation tables and failure summaries, reducing the human time needed to reconstruct experiment lineage before writing.

The assistants supported a range of repeatable project operations. They generated and revised Kaggle push folders, pushed inference kernels, downloaded public notebooks through the Kaggle CLI for inspection, packaged and uploaded Kaggle datasets, and recorded kernel logs and output paths after dry runs. For remote compute, they helped prepare or document vast.ai and Modal workflows used for training, caching, conversion, and benchmark jobs. We maintained a project runbook on Kaggle that captured recurring install, path-resolution, CUDA, ONNX, and OpenVINO issues, allowing later agent sessions to reuse known fixes instead of repeatedly debugging the same failures. A separate browser-based workflow, implemented with Vercel Labs' open-source `agent-browser` browser automation CLI [37], supported reconnaissance over Kaggle forums and public notebooks, including Kaggle forum pages that were not accessible to agents through the normal local or CLI-only workflow. Ideas found there were treated as leads for human-reviewed experiments, not as evidence by themselves.

These uses were limited to organization, implementation, and reproducibility support. The assistants could preserve context, flag missing metadata, and propose next checks; however, all final modeling choices, decisions to submit to the leaderboard, and scientific interpretation were reviewed by the human team.

## 5.2. Implementation and Artifact Verification

Beyond experiment tracking, we used coding agents as implementation support for repetitive but error-prone notebook and script work. They drafted or refactored helper code for Kaggle notebooks, model-conversion scripts, inference wrappers, and artifact checks, usually by adapting an existing working notebook rather than constructing a new pipeline from scratch. These tasks shared a common focus on defensive programming: ensuring that artifacts were discovered and loaded correctly, that class orderings were consistent across pipeline stages, and that model outputs satisfied basic sanity checks for shape, finite values, and non-degenerate prediction ranges.

The most important debugging role was mechanical verification of inference artifacts. We used the assistants to compare PyTorch, ONNX, and OpenVINO outputs after export; inspect frontend configuration such as mel dimensions and window shapes; and trace failures caused by missing metadata, missing attached sources, or path-resolution errors. These checks caught or documented issues that would otherwise be easy to overlook, such as frontend mismatches during ONNX export or notebook logic that silently loaded an unintended artifact. Where the assistants proposed and implemented fixes, the human team reviewed whether a corrected artifact or notebook was methodologically sound and appropriate for submission.

## 6. Conclusion

We presented a BirdCLEF+ 2026 system designed for low-resource passive acoustic monitoring under a strict CPU-only inference budget. The final ensemble combined OpenVINO SED streams with a Perch/ProtoSSM branch through per-class rank fusion, and outperformed every individual branch on both leaderboard splits. This result supports the central design choice: preserving complementary prediction sources was more valuable than optimizing only for the strongest standalone model.

Our experiments suggest that transfer signals must be used carefully in this setting. Perch was most reliable as a representation-level teacher through embedding distillation; direct output-space transfer from Perch was more fragile due to target-space mismatch. Pseudo labels contributed positively when they were sparse, positive-only, and low weight, but stronger pseudo-label pressure risked amplifying missing-label and teacher-error effects. Teacher diversity was beneficial when signals had distinct roles: Perch shaped acoustic representations, while CNN teachers shaped class-response rankings through teacher-logit cosine KD. Across these experiments, local OOF validation was useful for detecting broken runs and comparing close variants, but small OOF gains were not reliable evidence of hidden-test transfer.

The main limitation was the scarcity of labeled target-domain soundscapes. With only 66 labeled soundscape files, the available validation data could not support fine-grained recipe tuning or aggressive calibration. Future work should prioritize stronger target-domain validation protocols, uncertainty-aware pseudo-label selection, and calibration methods that preserve branch diversity without overfitting scarce labels. More broadly, our results show that successful BirdCLEF+ systems must treat modeling and deployment as a coupled problem: a high-performing acoustic model is useful only if it can also transfer across domains and run within the constraints of the target inference environment.

Our models learned to listen to the Pantanal. Whether they understood what they heard remains uncertain; the Pantanal, after all, has been speaking its own language long before we arrived with our spectrograms.

## Acknowledgements

We are grateful to the BirdCLEF+ 2026 organizers and to Kaggle for providing the competition setting, data, and evaluation platform that made this work possible. We also appreciate the participants who shared questions, implementation details, and experimental observations in the Kaggle discussion forums; those exchanges helped make the challenge a useful venue for bioacoustic monitoring research.

## Declaration on Generative AI

During the preparation of this work, the authors used Codex, Claude Code, and ChatGPT in order to aid programming, debugging, grammar and spelling checks, literature and forum review, and experiment documentation. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] World Wildlife Fund, The pantanal tropical wetland, https://www.worldwildlife.org/places/pantanal/, n.d. Accessed: 2026-06-05.

[2] The Nature Conservancy, The pantanal, https://www.nature.org/en-us/get-involved/how-to-help/places-we-protect/pantanal/, n.d. Accessed: 2026-06-05.

[3] W. M. Tomas, C. N. Berlinck, R. M. Chiaravalloti, G. P. Faggioni, C. Strüssmann, R. Libonati, C. R. Abrahão, G. do Valle Alvarenga, A. E. de Faria Bacellar, F. R. de Queiroz Batista, T. S. Bornato, et al., Distance sampling surveys reveal 17 million vertebrates directly killed by the 2020's wildfires in the pantanal, brazil, Scientific Reports 11 (2021) 23547. doi:10.1038/s41598-021-02844-5.

[4] L. S. M. Sugai, T. S. F. Silva, J. W. Ribeiro, D. Llusia, Terrestrial passive acoustic monitoring: Review and perspectives, BioScience 69 (2019) 15–25. doi:10.1093/biosci/biy147.

[5] 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.

[6] 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.

[7] E. Witting, H. de Heer, J. Lim, C. T. Kopar, K. Sándor, Addressing the challenges of domain shift in bird call classification for birdclef 2024, in: Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, volume 3740 of *CEUR Workshop Proceedings*, 2024. URL: https://ceur-ws.org/Vol-3740/paper-211.pdf.

[8] 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, volume 4038 of *CEUR Workshop Proceedings*, 2025. URL: https://ceur-ws.org/Vol-4038/paper_256.pdf.

[9] D.-H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on Challenges in Representation Learning, ICML, 2013. URL: https://www.researchgate.net/publication/280581078_Pseudo-Label_The_Simple_and_Efficient_Semi-Supervised_Learning_Method_for_Deep_Neural_Networks.

[10] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015. URL: https://arxiv.org/abs/1503.02531. arXiv:1503.02531.

[11] B. van Merriënboer, V. Dumoulin, J. Hamer, L. Harrell, A. Burns, T. Denton, Perch 2.0: The bittern lesson for bioacoustics, 2025. URL: https://arxiv.org/abs/2508.04665. doi:10.48550/arXiv.2508.04665. arXiv:2508.04665.

[12] Kaggle user mtoshidesu, Birdclef-2026, https://www.kaggle.com/code/mtoshidesu/birdclef-2026, 2026. Kaggle notebook. Accessed: 2026-06-05.

[13] OpenVINO Toolkit Contributors, Openvino, https://github.com/openvinotoolkit/openvino, 2026. GitHub repository. Accessed: 2026-06-11.

[14] Xeno-Canto Foundation, Xeno-canto: Sharing wildlife sounds from around the world, https://xeno-canto.org, n.d. Accessed: 2026-06-11.

[15] iNaturalist, inaturalist, https://www.inaturalist.org, n.d. Accessed: 2026-06-11.

[16] R. Wightman, Pytorch image models, https://github.com/huggingface/pytorch-image-models, 2019. GitHub repository. Accessed: 2026-06-11.

[17] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[18] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, P. Dollár, Designing network design spaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10428–10436. doi:10.1109/CVPR42600.2020.01044.

[19] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, S. Xie, Convnext v2: Co-designing and scaling convnets with masked autoencoders, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 16133–16142.

[20] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500. doi:10.1109/CVPR.2017.634.

[21] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141. doi:10.1109/CVPR.2018.00745.

[22] A. Brock, S. De, S. L. Smith, K. Simonyan, High-performance large-scale image recognition without normalization, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 1059–1071. URL: https://proceedings.mlr.press/v139/brock21a.html.

[23] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, Eca-net: Efficient channel attention for deep convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11534–11542. doi:10.1109/CVPR42600.2020.01155.

[24] S. Kahl, T. Denton, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF 2021: Bird call identification in soundscape recordings, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, volume 2936 of *CEUR Workshop Proceedings*, 2021, pp. 1437–1450. URL: https://ceur-ws.org/Vol-2936/paper-123.pdf.

[25] S. Kahl, A. Navine, T. Denton, H. Klinck, P. Hart, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF 2022: Endangered bird species recognition in soundscape recordings, in: Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, volume 3180 of *CEUR Workshop Proceedings*, 2022, pp. 1929–1939. URL: https://ceur-ws.org/Vol-3180/paper-154.pdf.

[26] S. Kahl, T. Denton, H. Klinck, H. Reers, F. Cherutich, 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*, 2023, pp. 1934–1942. URL: https://ceur-ws.org/Vol-3497/paper-164.pdf.

[27] S. Kahl, T. Denton, H. Klinck, V. Ramesh, V. Joshi, M. Srivathsa, A. Anand, C. Arvind, Harikrishnan CP, S. Sawant, Robin V V, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, 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, volume 3740 of *CEUR Workshop Proceedings*, 2024, pp. 1948–1957. URL: https://ceur-ws.org/Vol-3740/paper-184.pdf.

[28] 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*, 2025, pp. 2909–2919. URL: https://ceur-ws.org/Vol-4038/paper_232.pdf.

[29] N. Babych, birdclef2025 extra data(target and new species), https://www.kaggle.com/datasets/nikitababich/birdclef2025-1st-place-extra-data, 2025. Kaggle dataset. Accessed: 2026-06-06.

[30] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2008) 2579–2605. URL: https://www.jmlr.org/papers/v9/vandermaaten08a.html.

[31] ONNX Contributors, Onnx: Open neural network exchange, https://github.com/onnx/onnx, 2017. GitHub repository. Accessed: 2026-06-11.

[32] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, X. Wang, Metaformer baselines for vision, 2022. URL: https://arxiv.org/abs/2210.13452. arXiv:2210.13452.

[33] H. Pham, Z. Dai, Q. Xie, M.-T. Luong, Q. V. Le, Meta pseudo labels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11557–11568. URL: https://arxiv.org/abs/2003.10580.

[34] F. Radenović, G. Tolias, O. Chum, Fine-tuning cnn image retrieval with no human annotation, 2017. URL: https://arxiv.org/abs/1711.02512. arXiv:1711.02512.

[35] OpenAI, Codex, https://openai.com/codex/, 2026. Accessed: 2026-06-11.

[36] Anthropic, Claude code, https://code.claude.com/docs/en/overview, 2026. Accessed: 2026-06-11.

[37] Vercel Labs, agent-browser: Browser automation cli for ai agents, https://github.com/vercel-labs/agent-browser, 2026. GitHub repository. Accessed: 2026-06-09.

# A. Implementation Details

## A.1. Pseudo-Label Selection Details

**Table A1. Pseudo-label selection criteria used to construct the pseudo-positive label artifact.**

| Component | Requirement |
|---|---|
| HGNet logit-mean probability | >= 0.85 |
| HGNetV2-B6 probability | >= 0.80 |
| HGNetV2-B3 probability | >= 0.80 |
| SEResNeXt26t veto probability | >= 0.50 |

Selection was capped at 3 classes per window, 24 targets per file, 6 targets per file/class pair, and 800 targets per class. The generator processed 10,658 training soundscape files and produced 41,490 candidate positives before these caps.

**Table A2. Shared SED training configuration for models M1, M2, M3, and M4.**

| Item | Setting |
|---|---|
| Log-mel configuration | 32 kHz audio; 5 s crops; 256 x 313 log-mel spectrograms; `n_fft=2048`, `hop_length=512`, 256 mel bins, 20–16000 Hz, `top_db=80`; crop-wise z-score. |
| Training parameters | 25 epochs; batch size 16; AdamW with learning rate `5e-4` and weight decay `1e-4`; warmup-cosine schedule. |
| Supervised loss | Clip-level BCE averaged with BCE on maximum frame-level logits. |
| Distillation | Perch embedding MSE; cosine teacher–student KD with weight 0.3 where enabled. |
| Augmentation | Gain jitter, additive noise, focal-focal MixUp, focal-soundscape SumMix, and SpecAugment masks. |
| Data split | Stratified focal folds for local validation; labeled soundscape windows were included in training. |
| Code availability | Public repository: https://github.com/rashmibanthia/birdclef_2026. |

## A.2. Validation Shorthand

Several internal experiment notes report a non-S22 validation score. This was an auxiliary local OOF readout computed after excluding examples whose source-site code was S22. We used it only as a diagnostic for whether apparent local gains were concentrated in one site-specific slice of the small labeled soundscape set.

## A.3. Final Merger Post-Processing Details

Post-merge corrections were applied after fixed weighted rank fusion and before writing the final submission. The rules operated only on completed branch outputs and did not change any branch-internal calibration. They covered three cases: isolated high-confidence ProtoSSM responses with weak SED support, isolated high-rank SED responses with weak ProtoSSM support, and cases where multiple SED streams agreed while ProtoSSM was low.

Sonotype mirroring was applied after these corrections. For each group below, every member column was replaced by the row-wise maximum over that group:

```
47158son15, 47158son16
47158son09, 47158son12
47158son02, 47158son14
47158son13, 47158son21, 47158son22, 47158son23
```

Finally, rare non-bird classes in Amphibia, Mammalia, and Reptilia were lightly dampened below a class-specific threshold.