# Tackling Domain Shift in Bird Audio Classification via Transfer Learning and Semi-Supervised Distillation: A Case Study on BirdCLEF+ 2025

Volodymyr Sydorskyi[1,*], Fernando Gonçalves

[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

**Abstract**

We present our solution from team *volodymyr vialactea* to the BirdCLEF+ 2025 challenge, which achieved state-of-the-art performance, placing 2nd on the Private Leaderboard with a ROC AUC of 0.928 on the Private test set and 0.925 on the Public test set. Our system is based on five key components: a strong baseline model, in-domain transfer learning, semi-supervised learning implemented via model distillation to mitigate domain shift, postprocessing, and model ensembling. We conduct an ablation study to evaluate the contribution of each component and analyze the effects of different augmentations and data setups. Furthermore, we investigate the domain shift between training and test distributions and explore strategies for its mitigation. Our code is publicly available at https://github.com/VSydorskyy/BirdCLEF_2025_2nd_place.

**Keywords**

Birdcall classification, Transfer learning, Semi-supervised learning, Domain adaptation,

## 1. Introduction

Recent advancements in machine learning have demonstrated significant benefits across various domains of human activity, and ecological monitoring is no exception [1]. Machine and deep learning technologies have shown great promise in the field of bioacoustics [2]. The annual BirdCLEF series of competitions [3, 4, 5, 6, 7, 8] provides a vivid example of how the task of identifying bird species is both ecologically important and technically challenging. Moreover, the task continues to attract a growing number of participants—ranging from around 800 teams in earlier years to 2,025 teams in the BirdCLEF+ 2025 competition, which was organized as part of the LifeCLEF 2025 challenge [9, 10].

Participants in these competitions face several key challenges when building bird classification systems:

- Significant domain shift between recordings from passive acoustic monitoring (PAM) soundscapes and "directed" recordings from Xeno-Canto [11] and other databases;
- Severe class imbalance, with some species represented by as few as 2 recordings totaling just 11 seconds, while others have up to 1,218 recordings and more than 10 hours of audio;
- Large variability in recording devices and environmental conditions, both between training and test sets, and within the training recordings themselves;
- Computational constraints that require the system to run efficiently on microcomputers.

To address these challenges, we propose a solution based on the following core principles:

- Building a strong and reliable baseline model;
- Leveraging transfer learning by utilizing large-scale birdcall and birdsong databases;
- Applying domain adaptation via pseudo-labeling, which also serves as a form of model distillation;

*CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain*

*Corresponding author.

volodymyr.sydorskyi@gmail.com (V. Sydorskyi); fernando.eurico.goncalves@gmail.com (F. Gonçalves)

https://www.linkedin.com/in/volodymyr-sydorskyi (V. Sydorskyi); https://www.linkedin.com/in/fernandoeuricogoncalves (F. Gonçalves)

0000-0001-9697-7403 (V. Sydorskyi); 0009-0008-7132-1131 (F. Gonçalves)

- Ensembling diverse models, and performing post-processing of model probabilities.

The application of these techniques resulted in a highly robust and accurate system developed by team *volodymyr vialactea*, achieving a score of 0.928 on the Private Leaderboard and 0.925 on the Public Leaderboard—ultimately securing second place in the final Private ranking. In addition, we conduct an ablation study to evaluate the impact of individual components in our pipeline.

The remainder of this paper is organized as follows: Section 2 reviews related work that underpins our approach. Section 3 provides an overview of the BirdCLEF+ 2025 task. Section 4 details the structure and properties of the training and test datasets. Section 5 presents the proposed approach. Section 6 reports ablation studies and experimental findings. Finally, Section 7 discusses the main limitations of the current solution, outlines directions for future improvements, and concludes the paper.

## 2. Related Work

The gold standard in sound classification tasks is the use of a *Spectrogram → CNN → Classification head* architecture [12]. This pipeline is also widely adopted in birdcall classification tasks [13]. However, a range of modifications have been proposed. For example, [14] introduced a classification head that combines recurrent neural networks (RNNs) with an attention mechanism. This design enables training on weak (clip-level) labels while allowing the prediction of strong (frame-level) labels. One of the most widely used deep learning systems for birdcall classification is BirdNET [15]. Beyond the standard spectrogram-based CNN approach, BirdNET enhances its architecture by using multiple spectrogram variants that emphasize different frequency ranges. Alternative strategies include the use of 1D CNNs applied directly to raw audio signals [16], hybrid 1d CNN + 2d CNN methods [17], and transformer-based models such as ECAPA-TDNN [18]. However, these alternative methods have not consistently demonstrated superior performance in bird classification and tend to provide meaningful improvements primarily when used as part of ensembles—at least as observed in BirdCLEF competitions. A notable limitation of the BirdNET approach is its dependence on a fixed CNN backbone, specifically EfficientNet [19]. As newer CNN architectures such as NFNet [20] and EfficientNetV2 [21] are developed, BirdNET requires continuous architectural updates to remain competitive with state-of-the-art performance.

The challenge of domain adaptation is common across many machine learning tasks, and numerous methods have been proposed to address it [22, 23, 24]. However, domain-specific challenges often require tailored solutions. In the BirdCLEF 2024 competition, several approaches were proposed to address the domain shift between the focal (training) and soundscape (test) recordings. For instance, [25] applied classical semi-supervised learning to include unlabeled test examples in the training pipeline. Additionally, techniques such as no-call classifiers, test-time audio scaling, frequency-based noise removal, and domain-distance-based filtering were explored. It is important to note that since BirdCLEF 2024, unlabeled soundscape recordings have been made available to participants, enabling the use of semi-supervised learning techniques [26, 27]. Among the proposed solutions, using binarized pseudo-labels [25] tends to result in overconfident predictions, while hand-crafted domain adaptation strategies often lack flexibility and fail to introduce robust adaptation mechanisms. Thus, there is clear room for improvement in this direction.

In modern deep learning, transfer learning has become a foundational component across nearly all application domains [28]. In the audio classification domain, the situation is more nuanced. For example, PANNs—a family of CNNs pre-trained on the large-scale AudioSet—have shown strong performance across downstream audio tasks [29]. Interestingly, CNN encoders pre-trained on ImageNet also improve birdcall classification performance when applied to spectrograms [30]. Furthermore, recent approaches such as [25] utilize BirdNET and the Google Bird Vocalization Classifier [31] to extract audio embeddings and train dataset-specific classification heads on top. While these strategies clearly enhance performance, they still underutilize the full potential of transfer learning—especially in the context of continually expanding birdcall databases. This suggests that more aggressive and systematic use of transfer learning could yield further improvements.

# 3. Task Overview

**Table 1**
Distribution of Samples Across Animal Taxonomic Classes

| Taxonomic Class | Frequency |
|---|---|
| Aves | 146 |
| Amphibia | 34 |
| Insecta | 17 |
| Mammalia | 9 |

The BirdCLEF+ 2025 competition focused on the challenge of developing machine learning systems to identify under-studied species in the lowlands of the Magdalena Valley, Colombia, based on their acoustic signatures. Compared to previous editions, the competition expanded beyond bird species (*Aves*) to include other animal taxa as well (see Table 1). The training data were compiled from three major sources: Xeno-Canto [11], iNaturalist [32], and the CSA collection provided by the Instituto Humboldt [33]. The test data consisted of 1-minute long soundscape recordings collected through passive acoustic monitoring. All recordings were provided in Ogg format and resampled to 32 kHz. Participants were required to predict the probability of presence for each of the 206 target classes in every 5-second chunk of each audio file. The test set contained 700 soundscape samples, with a 34%/66% split between the Public and Private leaderboard sets. Additionally, strict computational constraints were imposed: all test-time predictions had to be executed within a 90-minute time limit using a Kaggle Notebook running on CPU-only (typically an Intel® Xeon® CPU @ 2.20GHz). Submissions were evaluated using a version of macro-averaged ROC-AUC that skips classes which have no true positive labels[1].

# 4. Data Overview

## 4.1. Training Data

**Table 2**
Distribution of Training Samples by Collection Source

| **Collection** | **Count** | **Percentage** |
|---|---|---|
| Xeno-Canto (XC) | 28,670 | 79.57% |
| iNaturalist (iNat) | 7,198 | 19.98% |
| CSA | 162 | 0.45% |

The training set consists of 28,564 audio samples, totaling approximately 280.5 hours of audio. The distribution across different data sources is highly imbalanced, with the vast majority of recordings coming from Xeno-Canto (see Table 2). A similar imbalance is observed in species distribution (see Figure 1): there are 39 species with fewer than 10 recordings—referred to as "undersampled"—and 9 species with more than 500 recordings. Overall, the imbalance can be quantified as Imbalance Ratio $= \frac{\max_i c_i}{\min_i c_i} = 495$, where $c_i$ is the sample count for class $i$. In addition to the primary species, some Xeno-Canto recordings include secondary (background) species, while such labels are absent in other sources. These secondary species appear in 4,958 recordings and can be used to model the multilabel structure of the test data. Finally, the training data was collected by 2,772 unique recordists, with 1,612 recordings lacking any metadata about the recordist.

[1] https://www.kaggle.com/code/metric/birdclef-roc-auc

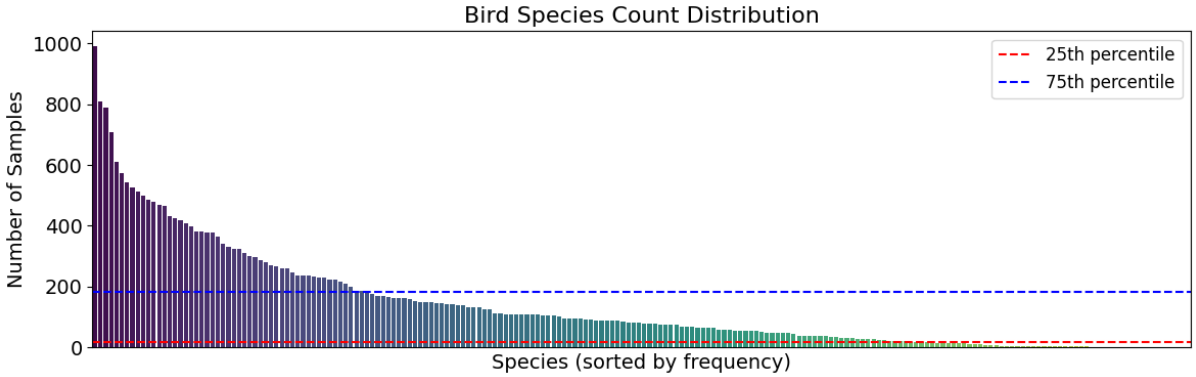


**Figure 1:** Counts of training samples per species (X-axis: species sorted by frequency; Y-axis: number of samples). Dashed lines indicate the 25th and 75th percentiles of the distribution.

## 4.2. Soundscapes

At the time of writing, the test set is not publicly available, preventing direct exploratory data analysis of test soundscapes. However, assuming that the test soundscapes follow the same (or at least a similar) distribution as the available unlabeled soundscapes, we approximate our analysis using pseudo labels (see Section 5.4). It is important to note that the following analysis may be affected by classification model bias, primarily inherited from the training data. For the analysis, we utilize pseudo labels from the first pseudo-iteration generated in non-OOF mode. In total, 9,726 unlabeled soundscape recordings are available, which correspond to 116,712 audio chunks of 5 seconds each. A primary point of interest is the analysis of domain shift between the training data and the soundscapes. We begin with a comparison of species distributions in the training data and the soundscapes. For this purpose, we count a species as present in a chunk if its predicted probability exceeds 0.5. As shown in Table 9, certain species are highly underrepresented in the training data but frequently appear in the soundscape recordings, and vice versa. Overall, the Pearson correlation between the number of training and soundscape occurrences per species is 0.0958, and the Spearman correlation is 0.2855. Another manifestation of domain shift lies in the number of species present in a single recording. In the training data, over 90% of samples contain only a single species, and the maximum number of species per recording is 12 (observed in only 2 recordings). In contrast, soundscape files demonstrate a much broader distribution: approximately 54% of recordings contain no identifiable species, 11% contain one species, 7% contain two, and some contain up to 25 species (see Table 10). Notably, soundscape files also include "nocall" samples. Another significant contributor to domain shift is the variation in recording conditions and devices. These differences are clearly illustrated by the spectrogram comparison shown in Figure 6. Finally, perhaps the most impactful aspect of domain shift lies in the annotation procedure: training data is annotated in a weak manner—labels apply to the entire recording, which may span several minutes, even though the bird vocalization may occur only for several seconds. In contrast, soundscape data is strongly labeled, providing species presence annotations at a fine temporal resolution of every 5-second chunk.

## 4.3. Additional Training Data

Using the open APIs of Xeno-Canto, iNaturalist, and CSA, we collected additional training samples for species of interest. We downloaded only the samples that comply with the competition's data license[2][3][4]. The full distribution of these additional samples is shown in Table 3. Interestingly, the most significant performance improvements came from incorporating the *Xeno-Canto dump from 28.03.2025* and a small set of *New data samples from previous competitions*. Including all additional datasets reduced the number of undersampled classes to 28, while using only the two most impactful sources reduced it to 38.

[2]https://www.kaggle.com/competitions/birdclef-2025/rules#7.-data-access-and-use
[3]https://www.kaggle.com/competitions/birdclef-2025/discussion/570760#3174229
[4]https://www.kaggle.com/competitions/birdclef-2025/rules#6.-external-data-and-tools

**Table 3**
Composition of the training dataset by data source, showing the number of audio samples contributed by each collection. Sources highlighted in bold were found to improve model performance.

| Data Source | Number of Samples |
|---|---|
| **Train Audio 2025** | **28,564** |
| **Xeno Canto dump from 28.03.2025** | **7,376** |
| INaturalist dump from 26.05.2025 | 6,482 |
| CSA dump from 11.05.2025 | 3,004 |
| Xeno Canto dump from 11.05.2025 | 170 |
| **New data samples from previous competitions** | **90** |

# 5. Method

## 5.1. Validation

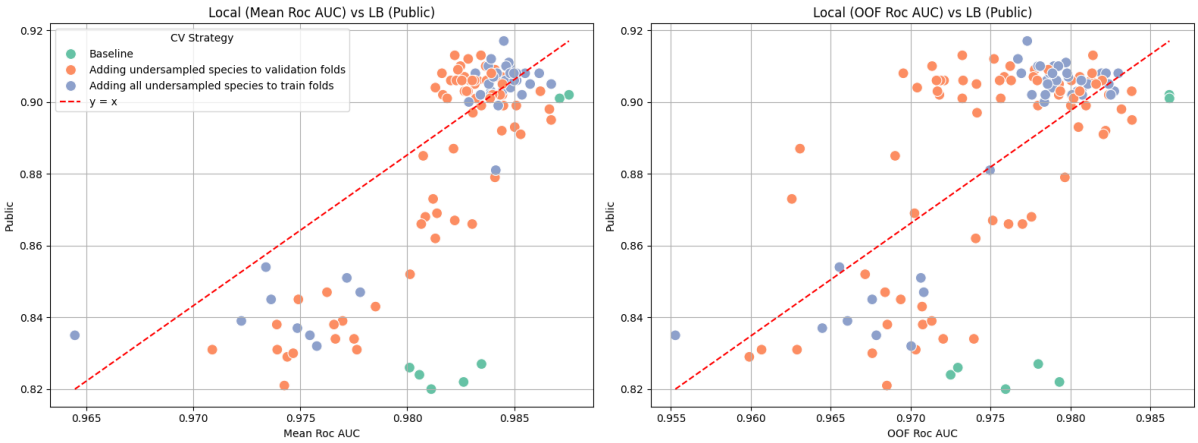


**Figure 2:** Scatter plots comparing local validation scores to Public Leaderboard scores across experiments and cross-validation strategies. The left plot shows Mean ROC AUC per fold versus Public ROC AUC, and the right plot shows OOF ROC AUC versus Public ROC AUC. Each point represents a single experiment, colored by cross-validation strategy. The red dashed line indicates the ideal correlation ($y = x$).

Building reliable validation remained one of the most significant challenges in the BirdCLEF series, primarily due to the domain shift described in Section 4.2. A baseline strategy was implemented using 5-fold cross-validation, stratified by primary species and grouped by author. This configuration helped maintain class distribution across folds while reducing potential data leakage, particularly from recordists contributing multiple sequential recordings. However, this baseline setup struggled with undersampled species, which were often represented by only 1–2 samples per fold or omitted entirely. To address this, the baseline was extended with one of the following modifications:

1. **Adding undersampled species to validation folds** – Undersampled species were placed in the validation set by ensuring at least one instance from each class, with corresponding samples removed from training. This allowed performance assessment on all classes, but further decreased the training data for rare species.
2. **Adding all undersampled species to train folds** – All available samples from undersampled species were included solely in the training set and excluded from validation. This eliminated evaluation of these classes but likely improved their recognition performance by enlarging the training data and minimizing label noise.

Despite these strategies, the correlation between local validation and Public Leaderboard scores remained relatively low. In general, large improvements in both validation and test performance were

observed during early experimentation. However, once the Public ROC AUC score approached 0.9, the correlation between validation metrics and test results weakened considerably. Two metrics were computed: Mean ROC AUC (averaged across all folds) and Out-of-Fold (OOF) ROC AUC (evaluated by concatenating all validation predictions). The Pearson and Spearman correlations between Mean ROC AUC and Public scores were $0.8643$ and $0.5438$, respectively. In contrast, within the high-performing region where Public scores exceeded $0.9$, correlations dropped sharply to $-0.1291$ (Pearson) and $-0.1200$ (Spearman). The corresponding scatter plots are shown in Figure 2.

Additionally, we experimented with mixing multiple recordings from the training set or even from soundscapes during the validation stage in an attempt to improve the consistency of validation scores. However, this approach did not increase the correlation between local validation and Public Leaderboard, and also introduced additional randomness due to varying sample combinations.

It is noteworthy that, as shown in Table 5, Public scores consistently exceeded $0.9$ once pseudo-labeling (see Section 5.4) was applied—serving primarily as a domain adaptation strategy. This observation suggests that validating directly on soundscape files may be a more appropriate approach. Finally, validation remained reasonably consistent during the final ensembling stage (see Section 6.2).

## 5.2. Baseline

The baseline approach was largely inspired by the BirdCLEF 2023 1st place solution[5]. The main distinction lies in replacing the ConvNeXt [34] model family with EfficientNetV2 [21]. Our approach primarily relied on two CNN backbones: EfficientNetV2-S and NFNet-L0. The key components of the baseline system included:

- **Augmentations to address domain shift:**
  - **MixUp** [35]: Two audio waveforms were added in the audio domain, and the resulting one-hot target vector was computed as the element-wise maximum across the original vectors. This augmentation was critical to better mimic the multilabel nature of the test data.
  - **Background mixing**: Background audio from prior-year soundscapes and the ESC-50 dataset [36] was randomly overlapped with training samples.
  - **SpecAugment** [37]: Standard time and frequency masking were applied to mel spectrograms.
  - **RandomFiltering**: A simplified version of a random equalizer was used to simulate channel distortions.
- **Data sampling strategy** followed Equation 1 with $\gamma = -0.5$.
- **Use of secondary labels** (if present) with equal weight to the primary label. This transformed the task into a multilabel classification problem, optimized via a linear combination of Binary Cross Entropy (BCE) and Focal loss [38].
- **Backbone and classification head**: An NFNet-L0 or EfficientNetV2-S CNN backbone was combined with a classification head inspired by [14], omitting RNN blocks and using weakly labeled prediction during both training and inference.
- **Additional datasets** were included as described in Section 4.3.

$$w_i = \left( \frac{c_i}{\sum_j c_j} \right)^{\gamma} \tag{1}$$

**where:**

- $w_i$ is the computed weight for class $i$,

[5] https://github.com/VSydorskyy/BirdCLEF_2023_1st_place

- $c_i$ is the number of samples for class $i$,
- $\gamma$ is the scaling exponent (typically $\gamma < 0$ to emphasize rare classes; $\gamma = -1$ corresponds to balanced sampling).

Models were trained for each validation fold, and final predictions on the Public and Private test sets were obtained by ensembling the best model from each fold via arithmetic averaging.

The baseline was enhanced by several modifications:

- **Label Smoothing** [39], as defined in Equation 2, with $\alpha = 0.05$.
- **Modified data sampling**, using two different strategies:
  1. For NFNet: Initial data sampling was used, but all classes with fewer than 100 samples were duplicated until they reached this threshold.
  2. For EfficientNetV2: Fully balanced sampling with $\gamma = -1$.

$$\tilde{y}_i = y_i \cdot (1 - \alpha) + \alpha \cdot \frac{\sum_j y_j}{K} \tag{2}$$

**where:**

- $\tilde{y}_i$ is the smoothed label for class $i$,
- $y_i$ is the original one-hot encoded label (1 for the correct class, 0 otherwise),
- $\alpha$ is the label smoothing factor,
- $K$ is the total number of classes.

To summarize, augmentations were introduced to mitigate domain shift and increase robustness to varying recording conditions, while also helping reduce overfitting to training data distribution. Data sampling strategies addressed the severe class imbalance and the evaluation metric's equal weighting across classes. Label smoothing was adopted to counter the noisy nature of weak labels. Finally, the chosen CNN backbones demonstrated strong performance in birdcall classification tasks and produced sufficiently diverse outputs, contributing positively to ensemble performance. The detailed training and inference setup is provided in Appendix B.

### 5.3. Transfer Learning

As mentioned in Section 2, CNN backbones pretrained on ImageNet already provide a strong initialization for birdcall classification. However, pretraining on in-domain data is expected to yield better performance. The pretraining pipeline was organized as follows:

1. **Species Selection**: A taxonomy of all species from previous BirdCLEF competitions was collected, resulting in a set of 16,607 unique species.
2. **Data Collection**: Audio recordings and associated metadata were downloaded from Xeno-Canto and from previous BirdCLEF competitions.
3. **Data Pruning**: To avoid data leakage, all species included in BirdCLEF+ 2025 were removed. Corrupted files and recordings with unmatched or invalid ebird codes were excluded. Additionally, species with fewer than 10 recordings were discarded to minimize label noise. The final pretraining dataset comprised 819,032 recordings spanning 7,489 species.
4. **Training Preparation**: The dataset was split into training and holdout subsets, with 5% of the data reserved for holdout evaluation. Validation was performed only on species with at least 100 recordings, to reduce evaluation noise. In total, the evaluation covered 1,627 bird species.
5. **Pretraining**: Models were trained using the baseline configuration (see Section 5.2), but without any class balancing. The objective was to learn general audio and birdcall structure, making sampling adjustments unnecessary at this stage.

6. **Checkpoint Preparation**: The best checkpoint based on holdout macro ROC AUC was selected. Only the CNN backbone was used; the classification head was reinitialized.
7. **Finetuning**: Fine-tuning was conducted using the same setup as described in Section 5.2, without any modifications to learning rate or scheduling.

Using a pretrained CNN backbone resulted in faster convergence during early training and consistently improved performance on both local cross-validation and the Public and Private test sets.

Additional experiments were conducted using an extended pretraining dataset that included samples from CSA, later Xeno-Canto dumps, and public Kaggle datasets. However, these variants did not yield further improvements in target metrics.

Moreover, several metric learning approaches were explored:

- Taking two disjoint 5-second segments from the same or different soundscape recordings and predicting whether they originated from the same audio file.
- Applying the previous strategy to training data recordings.
- Sampling two 5-second segments from the same or different species and predicting whether the species labels matched.

Unfortunately, none of these approaches proved effective. Due to the weak labels in the training data and noise in the pseudo labels for soundscapes, a substantial portion of many recordings contained background noise or mislabeled segments, which introduced too much noise for the models.

## 5.4. Pseudo Labelling

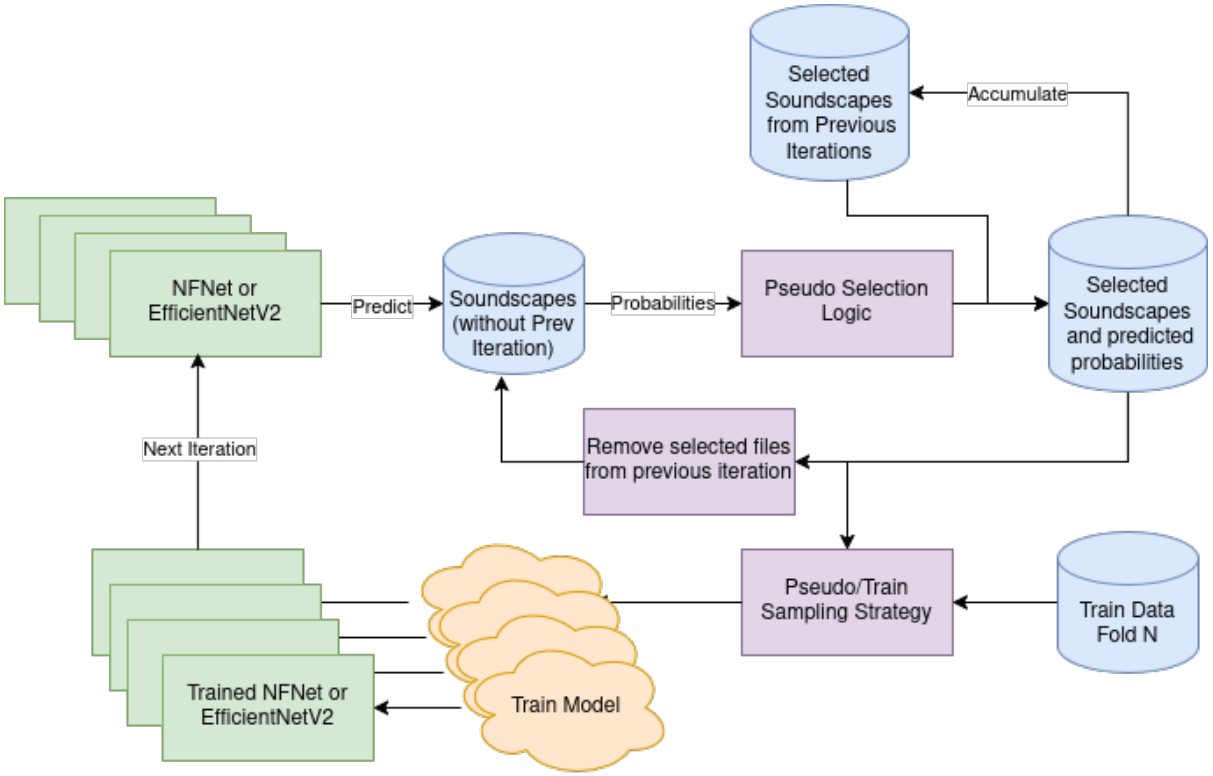


**Figure 3:** Pipeline for full soundscape pseudo-labeling. In each iteration, predictions are made on the full soundscape set (excluding previously selected files). Pseudo-labeled samples are accumulated across iterations, while previously selected samples are removed before the next iteration.

To better address domain shift and increase the volume of training data, a semi-supervised learning strategy was adopted. The overall pipeline is illustrated in Figure 3.

Initially, eight models were trained using the enhanced baseline configuration, comprising NFNet-L0 and EfficientNetV2-S backbones initialized from our pretrained checkpoints. These models differed in spectrogram types and optimization parameters. Their predictions on the unlabeled soundscape dataset were averaged via mean ensembling. Subsequently, a dedicated **Pseudo Selection Logic** was applied. For each 5-second chunk, the maximum predicted class probability was computed. Chunks with a maximum probability below $0.5$ were discarded. For retained chunks, all class probabilities below $0.1$ were zeroed out. This resulted in a pseudo-labeled dataset consisting of 5-second audio segments, each associated with a filtered soft target vector. The use of soft targets served two purposes: it discouraged overconfident predictions and enabled a form of knowledge distillation from the ensemble. Zeroing out low-confidence probabilities helped suppress noise from uncertain predictions.

Since the sampling strategy for the original training data was already carefully calibrated, we did not simply concatenate the pseudo-labeled dataset to the fold's training data. Instead, a dedicated **Sampling Strategy** was introduced, illustrated in Figure 4.

Additionally, the use of MixUp augmentation introduced an interesting fusion mechanism between hard labels from the original training data and soft labels from the pseudo-labeled samples. Audio segments were mixed in the audio domain, effectively creating an interpolated feature space. Corresponding label vectors were combined via element-wise summation and clipped to the $[0, 1]$ range. As a result, the final targets could simultaneously encode both soft and hard class probabilities.

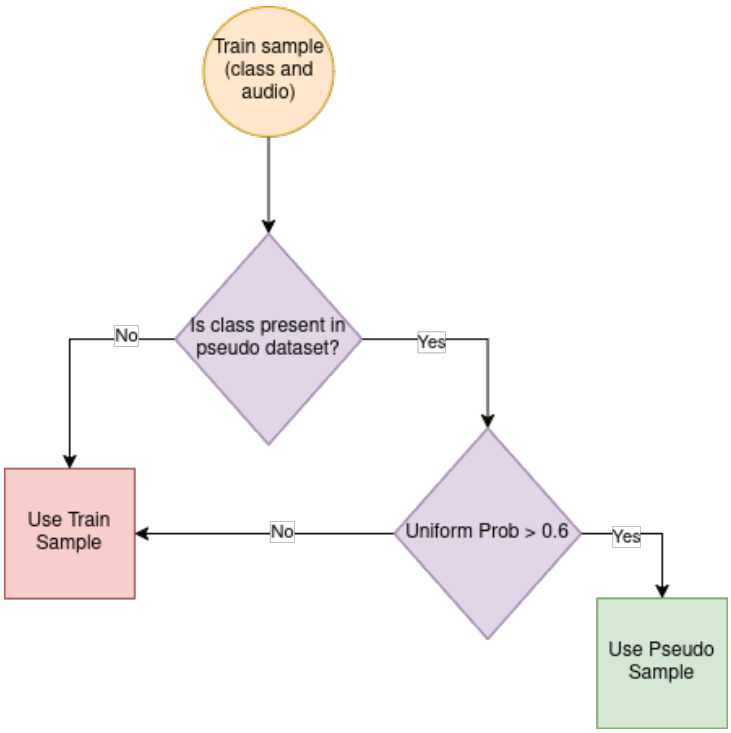


**Figure 4:** Sampling strategy for combining original training samples with pseudo-labeled soundscape data. For each training sample, if the corresponding class exists in the pseudo dataset, it is selected with 40% probability; otherwise, the original training sample is used.

In subsequent pseudo-labeling iterations, models were trained on the composition of previously selected pseudo samples. We employed both NFNet-L0 and EfficientNetV2-S architectures, again using pretrained initializations and enhanced baseline configurations. Each training experiment consisted of five folds, resulting in ten models (five per architecture) used to generate predictions for the next pseudo-labeling round. The number of selected samples per iteration is shown in Table 4. Described iterative pseudo-labeling algorithm is shown in Figure 3

One limitation of this iterative pseudo-labeling scheme is that samples from early iterations—predicted by weaker models—remain fixed in subsequent iterations. To mitigate this, we explored an alternative approach in which soundscapes were split by fold, and each fold was predicted using models not trained

**Table 4**
Statistics of selected pseudo-labeled soundscape samples across iterations. Note that models for each iteration were trained on cumulative pseudo-labeled data from all previous and current iterations.

| Pseudo Iteration | Selected Files | Selected Chunks |
|---|---|---|
| Iteration 1 | 4430 | 19,405 |
| Iteration 2 | 1483 | 3,750 |
| Iteration 3 | 1437 | 4,108 |

on it (see Figure 5). This out-of-fold (OOF) strategy enabled refreshing pseudo-labels at every iteration and increased model diversity due to varying predictions across folds. However, this OOF approach comes with trade-offs: each 5-second chunk is predicted by only two models instead of ten, reducing ensemble stability, and each model is trained on only $\frac{4}{5}$ of the pseudo-labeled dataset, rather than the full set.

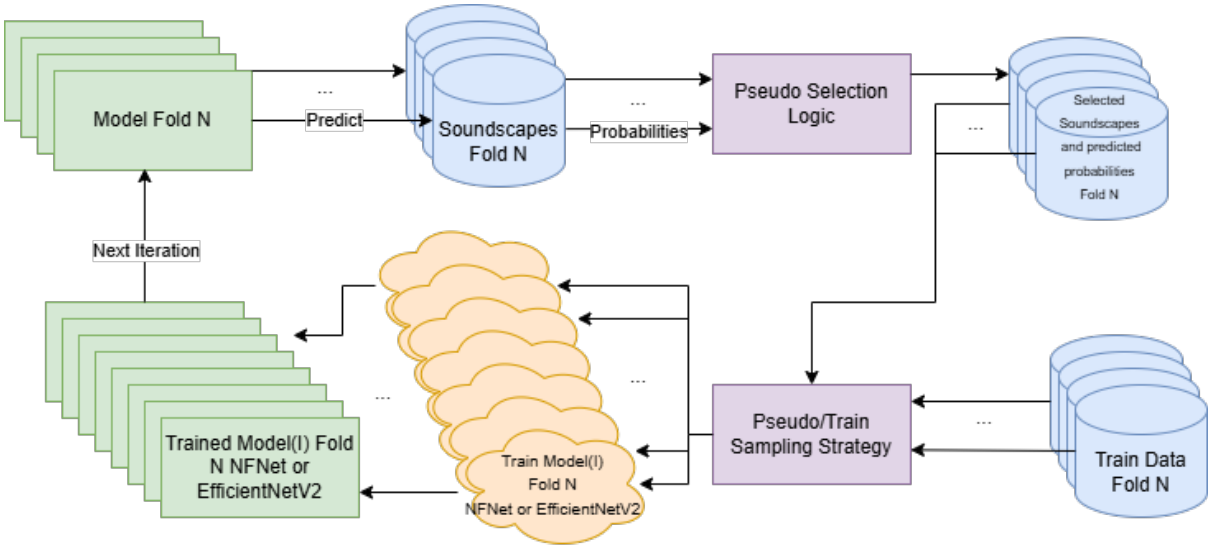


**Figure 5:** Overview of the out-of-fold (OOF) pseudo-labeling pipeline. In this setup, all soundscape files are predicted at every iteration. The soundscapes are split by fold, and each fold is predicted using models not trained on that fold, producing updated pseudo-labels across iterations.

Finally, pseudo-labeled samples from both strategies were utilized independently and also merged into a single dataset using the same "removal from previous iteration" logic (see Figure 3). Training models on different subsets of pseudo data contributed to ensemble diversity and improved generalization on the final leaderboard submissions.

## 5.5. Data Curation

We trained our models on 5-second segments randomly selected from the input audio. Based on past experience, we initially tried three approaches for picking 5-second segments: random 5 seconds from the whole audio, random 5 seconds from the first 7 seconds, and random 5 seconds from the first or last 7 seconds. The reasoning for the last two approaches is that recordings are often started when the target animal is already vocalizing and stopped when it ceases. This can help the model reduce false positives by focusing on parts of the audio more likely to contain vocalizations. The 7-second window was chosen to introduce variability while still capturing likely vocal activity.

In initial tests, the last approach provided better results. However, some species had only a few audios - as few as 2 in some cases - and we were concerned about overfitting. Therefore, we decided to inspect the audios for those species to manually identify sections of vocalization. Three such cases are presented below:

1. **Vocalization with alien speech** (Figure 7). Several audios contained a computer-synthesized voice with an ID for the recording. Some audios also included a section in which the recordist describes the audio, e.g., species recorded, location, temperature, microphone used, etc. In many of these cases, the description contains no traces of vocalization or even the same background noise as the sections in which there is vocalization. We refer to these two instances of human voice as alien speech. For some audios, alien speech represented more than 90% of the recording.
2. **Speech overlapping animal vocalization** (Figure 8). In this case, the voice can be understood as background noise.
3. **Vocalization with periods of silence** (Figure 9), i.e., when the animal is not vocalizing.

All these cases would result in training with false positives, with the potential to hinder learning. Hence, we eliminated those sections from our training. However, that ended up hurting our models. The reasons were not entirely clear. It is possible that our audio curation was overly aggressive and excluded some true positives. Alternatively, the presence of some false positives may have acted as a form of regularization, improving generalization by reducing overfitting to the limited number of available recordings.

We also experimented with extensive manual curation for species with fewer than 30 samples, listening to the recordings while simultaneously inspecting spectrograms and energy level charts. As this approach was not scalable for larger classes, we additionally applied automatic speech detection and made several attempts to adjust labels based on predictions from our strongest models. The intuition was that if a model consistently assigned low probabilities to a label, that label was likely incorrect for that segment; similarly, consistently high probabilities for a species not in the labels might indicate an overlooked species. We tried using these predictions as soft labels, hard labels, or to cancel the original annotations when probabilities were very low.

Unfortunately, none of these curation strategies led to consistent performance improvements. In the end, we decided to either use the whole audio or exclude only the sections of alien speech identified manually or automatically.

### 5.6. Postprocessing

We attempted several forms of post-processing, which were based on two hypotheses:

- If an animal is vocalizing in an audio recording, it will appear in multiple segments.
- The vocalizations of some species follow rhythmic patterns. For example, an animal may vocalize once every second for 16 seconds, then pause for 6 seconds, and repeat this cycle.

From the first hypothesis, we derived the following methods:

- **Mean**: multiply the probability of a species in each segment by the average probability of the species in the whole audio. The intuition is that if a species has a high probability in other segments, a high probability in a given segment is more trustworthy when compared to other audios. The converse is also true.

$$\text{postproc\_prob}_{a,s,c} = \text{prob}_{a,s,c} \cdot \left( \frac{1}{S} \sum_{s'=1}^{S} \text{prob}_{a,s',c} \right) \tag{3}$$

  **where:**

  - $a$ – audio index,
  - $s$ – segment index (5-second chunk),
  - $c$ – species class index,
  - $S$ – total number of segments in audio $a$.

- **TopN**: multiply the probability of a species in each segment by the average top N probabilities of the species in the whole audio. The intuition for this approach was similar to **Mean**, with the additional intention of ignoring periods of silence in which the species is not vocalizing.

$$\text{postproc\_prob}_{a,s,c} = \text{prob}_{a,s,c} \cdot \left( \frac{1}{N} \sum_{s' \in \mathcal{T}_{a,c}} \text{prob}_{a,s',c} \right) \tag{4}$$

  **where:**
  - $\mathcal{T}_{a,c}$ – the set of indices for the top $N$ values of $\text{prob}(a, :, c)$.
  - $N$ – the number of top segments used for smoothing the probability.
- **Convolution**: adjust the probability of each segment by applying a convolution of the probabilities of neighboring segments. The intuition is that if a species has a high probability in neighboring segments, it is more likely to be vocalizing in a given segment.

From the second hypothesis, we derived the following method:

- **L2 model**: A layer 2 model (L2) was trained to refine the predictions for each test audio. The intuition behind this approach is that if a model can learn the rhythmic patterns of a species' vocalization, it can correct errors from the layer 1 model (L1) and produce predictions that better reflect the species' natural behavior. The L2 model was trained on the pseudo-labeled (see Section 5.4) soundscape recordings using species-level vocalization probabilities for each 5-second segment. To simulate prediction errors from the L1 model, random noise was added to the original probabilities to create the inputs. The training labels were generated by thresholding the original probabilities at 0.5, converting them into hard labels. Training was restricted to species with at least 10 audio files. Several model architectures were evaluated, and a simple convolution model with different kernels per species yielded the best results.

Our tests showed that the first two methods improved the original predictions. Unfortunately, the **L2 model** did not yield improvements over the **TopN** method on the Public Leaderboard, even when the two were combined. Combining the other methods with each other also did not result in further gains. Our best results were consistently achieved using the **TopN** method with $N = 1$.

## 5.7. Ensembling

To obtain a robust final solution, an ensembling strategy was employed. As previously discussed, five models (one from each training fold) were used per experiment to generate predictions for the Public test set. In the final ensemble, we performed a simple average over the predictions of all five folds across three selected experiments, resulting in an ensemble of 15 models. Postprocessing was then applied to the averaged predictions.

To select the optimal experiments for inclusion in the ensemble, we explored two strategies:

- Selecting high-performing models based on their Public Leaderboard scores and maximizing the ensemble's Public score directly.
- Selecting three experiments using Optuna [40], with the objective of maximizing the OOF validation macro ROC AUC. Additionally, we tracked the individual contribution of each experiment by measuring the performance gain over the best single model.

The Optuna-based selection yielded the highest Private score, suggesting that even when validation metrics are only weakly correlated with test performance, they may still provide useful guidance for constructing effective ensembles.

Ranking-based ensembling was also explored; however, it did not lead to competitive performance. To enable compatibility with postprocessing, the pipeline applied postprocessing first, followed by rank transformation, and then arithmetic averaging. We hypothesize that our models are relatively well-calibrated, so their raw probability outputs contribute meaningfully to the final score. Moreover, applying postprocessing directly to individual models often resulted in degraded performance compared to applying it at the ensemble level.

# 6. Results

## 6.1. Main Results and Ablation Study

**Table 5**
Stepwise evaluation of model improvements. "I" denotes the pseudo-labeling iteration number ("I1" = Iteration 1). "Full" refers to full soundscape pseudo-labeling, "OOF" stands for out-of-fold pseudo-labeling, and "PP" indicates postprocessing applied after prediction.

| Incremental Change | Configuration | CV Strategy | Mean CV Score | Public | Private |
|---|---|---|---|---|---|
| **Baseline** | | | | | |
| | NFNet-L0 | 2nd | 0.978 | 0.847 | 0.868 |
| | EfficientNetV2-S | 2nd | 0.975 | 0.837 | 0.859 |
| + **Enhancements** | | | | | |
| | NFNet-L0 | 2nd | 0.977 | 0.851 | 0.875 |
| | EfficientNetV2-S | 2nd | 0.975 | 0.835 | 0.871 |
| + **Transfer Learning** | | | | | |
| | NFNet-L0 | 1st | 0.980 | 0.866 | 0.872 |
| | EfficientNetV2-S | 2nd | **0.984** | 0.881 | 0.889 |
| + **Pseudo Labels** | | | | | |
| | Full I1 (EffNetV2-S) | 2nd | **0.985** | 0.908 | 0.906 |
| | Full I2 (EffNetV2-S) | 2nd | **0.984** | 0.910 | 0.909 |
| | Full I3 (EffNetV2-S) | 2nd | **0.984** | 0.902 | 0.907 |
| | Full I3 (NFNet-L0) | 1st | 0.983 | 0.913 | 0.913 |
| | OOF I1 (EffNetV2-S) | 2nd | **0.985** | 0.906 | 0.908 |
| | OOF I2 (EffNetV2-S) | 2nd | **0.985** | 0.911 | 0.910 |
| | OOF I2 + Full I2 (EffNetV2-S) | 2nd | **0.985** | **0.917** | 0.910 |
| + **Postprocessing** | | | | | |
| | Full I2 + PP (EffNetV2-S) | — | — | **0.918** | **0.924** |
| | Full I3 + PP (NFNet-L0) | — | — | **0.917** | **0.924** |

The progression of model improvements is summarized in Table 5. The baseline models already demonstrated strong performance on the training data, although a substantial gap was observed between cross-validation (CV) and Public/Private test scores. NFNet-L0 showed a clear advantage over EfficientNetV2-S at this stage. Introducing the proposed enhancements (see Section 5.2) resulted in comparable CV performance, while yielding noticeable improvements in Private scores and modest gains on the Public set. This behavior aligns with expectations, as the enhancements were designed to handle noisy targets and solve the class imbalance problem.

The introduction of transfer learning further boosted all metrics, reinforcing the effectiveness of pretraining on large in-domain birdcall datasets. Interestingly, this stage reversed the earlier trend of NFNet-L0 superiority, with EfficientNetV2-S now outperforming it. This shift may be attributed to differences in optimization policies between the architectures and requires further investigation in future work.

The most substantial gains were achieved through pseudo-labeling, which significantly reduced the domain shift. Validation scores remained stable or slightly improved, suggesting that the pseudo-labeled data did not introduce excessive noise. In contrast, Public and Private scores improved markedly. Both full and OOF pseudo-labeling strategies benefited from a second iteration, while performance plateaued or slightly declined in the third, particularly for EfficientNetV2-S. This may indicate that the most informative (i.e., less ambiguous) segments had already been captured. No consistent preference emerged between full and OOF pseudo-labeling strategies.

Finally, postprocessing (**TopN** method with $N = 1$) contributed an additional 1–1.5% improvement in ROC AUC on the Public and Private sets, further validating our hypothesis from Section 5.6 regarding

the repeated vocalizations of birds across the same file. We do not report postprocessing scores on local validation, as 5-second chunk-level labels are not available for the training data.

## 6.2. Ensembling Results

**Table 6**
Comparison of different ensemble strategies. The "CV Gain Over Best Model" column shows the improvement in OOF ROC AUC compared to the highest-scoring single model included in each ensemble.

| Ensemble Strategy | OOF CV Score | CV Gain Over Best Model | Public | Private |
|---|---|---|---|---|
| Best Public ensemble | 0.992 | 0.0049 | **0.925** | 0.928 |
| Optuna optimized ensemble | **0.993** | **0.0057** | 0.921 | **0.929** |

Table 6 summarizes the results of different ensemble strategies. Ensembling results weakly indicate that relying on validation scores may be a more reliable strategy than selecting models solely based on Public leaderboard scores. The intuition behind this is that validation scores of ensembles behave differently from those of individual models, as they implicitly reflect the degree of model correlation and the ensemble's ability to compensate for errors made by individual models. However, due to the marginal score differences, the observed improvements may not be statistically significant. Notably, ensembling provided an average ROC AUC improvement of about 0.5% compared to the best individual models. While such a boost can help secure a Top 2 leaderboard position, it is relatively modest overall, highlighting that our single models already achieved strong performance. This characteristic makes them particularly well suited for production settings where inference speed and simplicity are critical.

## 6.3. Impact of Additional Training Data

**Table 7**
Evaluation results for different additional data setups.

| Data Setup | CNN Backbone | CV Strategy | Mean CV score | Public | Private |
|---|---|---|---|---|---|
| Only Train Data | EfficientNetV2-S | 2nd | 0.972 | **0.839** | 0.868 |
| Adding 2 selected additional datasets | EfficientNetV2-S | 2nd | 0.975 | 0.835 | **0.871** |
| Adding all additional datasets | EfficientNetV2-S | 2nd | **0.976** | 0.832 | 0.864 |

Table 7 presents the results of evaluating different additional data configurations. Compared to BirdCLEF 2023[6], in this year's competition the inclusion of additional data did not yield substantial performance improvements. In fact, incorporating large volumes of extra data slightly degraded results on both the Public and Private leaderboards. This behavior can be explained by the already sufficient size of the 2025 dataset, where enriching it with more samples brings limited benefit. Additionally, some of the external data sources contained systematic speech artifacts (see Section 5.5), which may have introduced false positives and confused the models. Still, we hypothesize that performance for certain undersampled species should be improved. A deeper analysis of this aspect is left for future research.

## 6.4. Augmentation Ablation

Table 8 shows the results of the augmentation ablation study. The augmentation ablation study demonstrates that while MixUp does not lead to improvements on the Public test set, it significantly boosts performance on the Private set and improves local validation scores. This suggests that MixUp contributes to training more robust models. We hypothesize that the absence of performance gain on the Public leaderboard may be due to a lower number of overlapping bird vocalizations per 5-second

[6] https://www.kaggle.com/competitions/birdclef-2023/discussion/412808

**Table 8**
Evaluation results for different augmentation setups.

| Augmentation Setup | CNN Backbone | CV Strategy | Mean CV score | Public | Private |
|---|---|---|---|---|---|
| Enhanced Baseline | EffNetV2-S | 2nd | **0.975** | 0.835 | **0.871** |
| Without SpecAugment | EffNetV2-S | 2nd | 0.974 | 0.845 | 0.861 |
| Without RandomFiltering | EffNetV2-S | 2nd | 0.973 | **0.854** | 0.861 |
| Without MixUp | EffNetV2-S | 2nd | 0.964 | 0.835 | 0.835 |

chunk compared to the Private set. SpecAugment and RandomFiltering slightly decrease Public scores but provide noticeable gains on the Private set and minor improvements in local validation. Overall, these results support the conclusion that applying heavier augmentations leads to more robust models, consistent with general theoretical expectations.

# 7. Limitations and Conclusions

## 7.1. Limitations

Our work has several limitations that should be acknowledged:

- **Lack of Robust Evaluation**: The local validation strategy does not reflect the real-world deployment scenario of the system. Furthermore, the available soundscapes are biased towards a single geographic location. As our approach relies heavily on pseudo labels derived from these data, the model's performance may degrade when applied to soundscapes from other regions.
- **Limited Hyperparameter Optimization**: The focus of this work was primarily on exploring methodological improvements rather than optimizing configurations. As a result, we did not perform an extensive hyperparameter search, especially for pseudo-labeling thresholds and model settings, which may have limited the achievable performance.

## 7.2. Conclusions

In this work, we proposed a system that achieved state-of-the-art performance on the BirdCLEF+ 2025 task, placing 2nd overall with a Public Leaderboard score of 0.925 and a Private Leaderboard score of 0.928. Our approach is built on five key pillars: a strong baseline model, transfer learning from large-scale birdcall datasets, semi-supervised learning with elements of model distillation, postprocessing, and ensembling. We further complemented our work with in-depth data analysis and ablation studies to better understand the contributions of each component.

Despite these achievements, there remains substantial room for future improvement. Potential directions include:

- Developing a more robust evaluation framework by retraining the same pipelines on different subsets of the data and evaluating them on soundscapes from diverse geographic locations. Alternatively, training a unified model on all available soundscapes and classes may lead to more generalizable performance.
- Employing model-driven filtering to identify and exclude non-vocalized audio fragments in the training set, thereby reducing label noise and improving optimization efficiency.
- Leveraging the full capabilities of strong label prediction models such as [14], particularly during inference, to better capture temporal bird vocalization patterns.
- Addressing the challenge of undersampled species.

## Acknowledgments

We would like to thank the organizers of BirdCLEF, the Kaggle team, and LifeCLEF for hosting this competition. We are especially grateful to the Armed Forces of Ukraine—without their resilience and protection, this work would not have been possible.

## Declaration on Generative AI

During the preparation of this work, the authors used OpenAI's ChatGPT in order to improve clarity, coherence, and LaTeX formatting. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] E. Alotaibi, N. Nassif, Artificial intelligence in environmental monitoring: in-depth analysis, Discover Artificial Intelligence 4 (2024) 84.

[2] D. Stowell, Computational bioacoustics with deep learning: a review and roadmap. peerj 10, e13152, 2022.

[3] A. Howard, H. Klinck, S. Dane, S. Kahl, tom denton, T. Denton, Cornell birdcall identification, https://kaggle.com/competitions/birdsong-recognition, 2020. Kaggle.

[4] A. Howard, A. Joly, H. Klinck, S. Dane, S. Kahl, tom denton, T. Denton, Birdclef 2021 - birdcall identification, https://kaggle.com/competitions/birdclef-2021, 2021. Kaggle.

[5] A. Howard, A. Navine, H. Klinck, S. Dane, S. Kahl, T. Denton, Birdclef 2022, https://kaggle.com/competitions/birdclef-2022, 2022. Kaggle.

[6] H. Klinck, S. Dane, S. Kahl, T. Denton, Birdclef 2023, https://kaggle.com/competitions/birdclef-2023, 2023. Kaggle.

[7] H. Klinck, Maggie, S. Dane, S. Kahl, T. Denton, V. Ramesh, Birdclef 2024, https://kaggle.com/competitions/birdclef-2024, 2024. Kaggle.

[8] H. Klinck, J. S. Cañas, M. Demkin, S. Dane, S. Kahl, T. Denton, Birdclef+ 2025, https://kaggle.com/competitions/birdclef-2025, 2025. Kaggle.

[9] L. Picek, S. Kahl, H. Goëau, L. Adam, et al., Overview of lifeclef 2025: Challenges on species presence prediction and identification, and individual animal identification, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2025.

[10] J. S. Cañas, S. Kahl, T. Denton, M. P. Toro-Gómez, S. Rodriguez-Buritica, J. L. Benavides-Lopez, J. S. Ulloa, P. Caycedo-Rosales, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF+ 2025: Multi-taxonomic sound identification in the middle magdalena valley, colombia, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, 2025.

[11] Xeno-Canto Foundation, Xeno-canto: Sharing bird sounds from around the world, https://xeno-canto.org/, 2025. Accessed: June 13, 2025.

[12] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, P. Tiwari, Sound classification using convolutional neural network and tensor deep stacking network, IEEE Access 7 (2019) 7717–7727.

[13] M. Lasseck, Acoustic bird detection with deep convolutional neural networks., in: DCASE, 2018, pp. 143–147.

[14] S. Adavanne, T. Virtanen, Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network, arXiv preprint arXiv:1710.02998 (2017).

[15] S. Kahl, C. M. Wood, M. Eibl, H. Klinck, Birdnet: A deep learning solution for avian diversity monitoring, Ecological Informatics 61 (2021) 101236.

[16] S. Abdoli, P. Cardinal, A. L. Koerich, End-to-end environmental sound classification using a 1d convolutional neural network, Expert Systems with Applications 136 (2019) 252–263.

[17] R. Kulkarni, B. Chandarana, Audio based species identification for monosyllabic call birds using convolutional neural networks, International Journal for Research in Applied Science & Engineering Technology (IJRASET) 7 (2019).
[18] B. Desplanques, J. Thienpondt, K. Demuynck, Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification, arXiv preprint arXiv:2005.07143 (2020).
[19] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.
[20] A. Brock, S. De, S. L. Smith, K. Simonyan, High-performance large-scale image recognition without normalization, in: International conference on machine learning, PMLR, 2021, pp. 1059–1071.
[21] M. Tan, Q. Le, Efficientnetv2: Smaller models and faster training, in: International conference on machine learning, PMLR, 2021, pp. 10096–10106.
[22] J. Li, Z. Yu, Z. Du, L. Zhu, H. T. Shen, A comprehensive survey on source-free domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).
[23] G. Wilson, D. J. Cook, A survey of unsupervised deep domain adaptation, ACM Transactions on Intelligent Systems and Technology (TIST) 11 (2020) 1–46.
[24] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR, 2015, pp. 1180–1189.
[25] A. Miyaguchi, A. Cheung, M. Gustineli, A. Kim, Transfer learning with pseudo multi-label birdcall classification for ds@ gt birdclef 2024, arXiv preprint arXiv:2407.06291 (2024).
[26] D.-H. Lee, et al., Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: Workshop on challenges in representation learning, ICML, volume 3, Atlanta, 2013, p. 896.
[27] P. Kage, J. C. Rothenberger, P. Andreadis, D. I. Diochnos, A review of pseudo-labeling for computer vision, arXiv preprint arXiv:2408.07221 (2024).
[28] J. Jiang, Y. Shu, J. Wang, M. Long, Transferability in deep learning: A survey, arXiv preprint arXiv:2201.05867 (2022).
[29] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, M. D. Plumbley, Panns: Large-scale pretrained audio neural networks for audio pattern recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020) 2880–2894.
[30] D. B. Efremova, M. Sankupellay, D. A. Konovalov, Data-efficient classification of birdcall through convolutional neural networks transfer learning, in: 2019 Digital image computing: Techniques and applications (DICTA), IEEE, 2019, pp. 1–8.
[31] B. Ghani, T. Denton, S. Kahl, H. Klinck, Global birdsong embeddings enable superior transfer learning for bioacoustic classification, Scientific Reports 13 (2023) 22876.
[32] California Academy of Sciences and the National Geographic Society, inaturalist, https://www.inaturalist.org/, 2025. Accessed: June 13, 2025.
[33] Visor de sonidos – colecciones humboldt, https://colecciones.humboldt.org.co/sonidos/visor-csa/, 2025. Accessed: June 13, 2025.
[34] 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.
[35] K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, S. Liu, Mixup-based acoustic scene classification using multi-channel convolutional neural network, in: Advances in Multimedia Information Processing–PCM 2018: 19th Pacific-Rim Conference on Multimedia, Hefei, China, September 21-22, 2018, Proceedings, Part III 19, Springer, 2018, pp. 14–23.
[36] K. J. Piczak, ESC: Dataset for Environmental Sound Classification, in: Proceedings of the 23rd Annual ACM Conference on Multimedia, ACM Press, ????, pp. 1015–1018. URL: http://dl.acm.org/citation.cfm?doid=2733373.2806390. doi:10.1145/2733373.2806390.
[37] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, Q. V. Le, Specaugment: A simple data augmentation method for automatic speech recognition, arXiv preprint arXiv:1904.08779 (2019).

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[39] R. Müller, S. Kornblith, G. E. Hinton, When does label smoothing help?, Advances in neural information processing systems 32 (2019).
[40] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.
[41] A. Paszke, Pytorch: An imperative style, high-performance deep learning library, arXiv preprint arXiv:1912.01703 (2019).
[42] W. Falcon, The PyTorch Lightning team, PyTorch Lightning, 2019. URL: https://github.com/Lightning-AI/lightning. doi:10.5281/zenodo.3828935.
[43] R. Wightman, Pytorch image models, https://github.com/rwightman/pytorch-image-models, 2019. doi:10.5281/zenodo.4414861.
[44] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).
[45] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, J. Han, On the variance of the adaptive learning rate and beyond, arXiv preprint arXiv:1908.03265 (2019).
[46] K. W. Cheuk, H. Anderson, K. Agres, D. Herremans, nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks, IEEE Access 8 (2020) 161981–162003.
[47] F. Radenović, G. Tolias, O. Chum, Fine-tuning cnn image retrieval with no human annotation, IEEE transactions on pattern analysis and machine intelligence 41 (2018) 1655–1668.
[48] A. Collette, Python and HDF5, O'Reilly, 2013.
[49] OpenVINO Toolkit Contributors, OpenVINO Toolkit, https://github.com/openvinotoolkit/openvino, 2024. Accessed: 2025-06-13.

## A. Data Tables and Figures

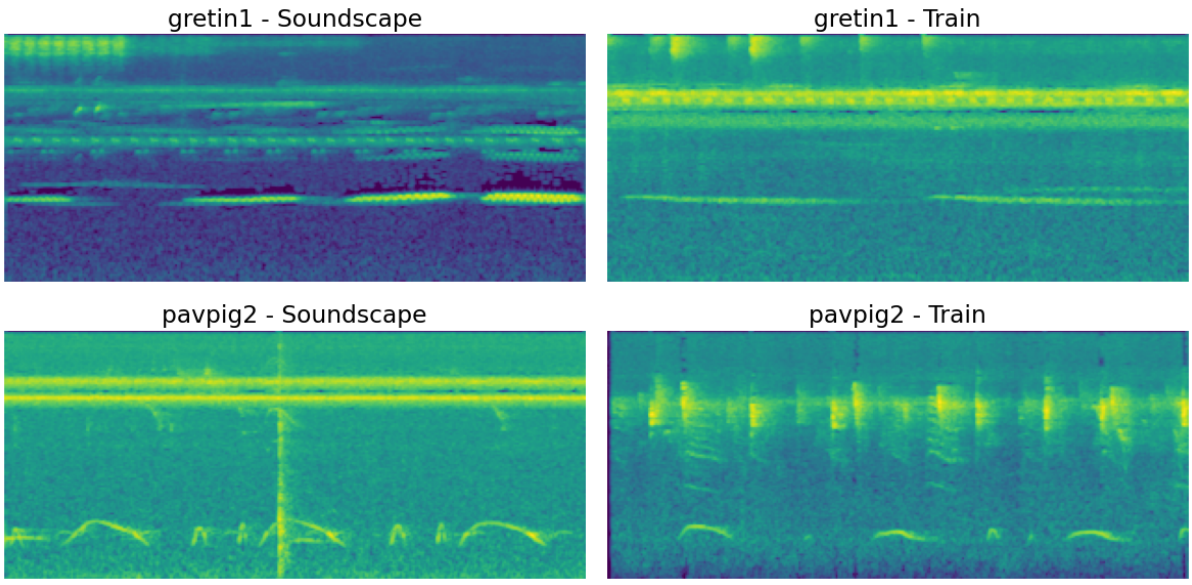


**Figure 6:** Spectrogram examples for two bird species (*gretin1* and *pavpig2*) in both training and soundscape datasets

**Table 9**
Bird Species with Highest and Lowest Soundscape Occurrence

| Top 10 by Soundscape Count | Train Count | Soundscape Count |
|---|---|---|
| pavpig2 | 259 | 1543 |
| 52884 | 33 | 1200 |
| 555142 | 6 | 870 |
| 64862 | 3 | 862 |
| blbwre1 | 184 | 799 |
| 22976 | 46 | 776 |
| 65448 | 84 | 729 |
| blbgra1 | 560 | 613 |
| littin1 | 502 | 591 |
| rugdov | 288 | 588 |
| **Only 1 Occurrence in Soundscapes** | **Train Count** | **Soundscape Count** |
| whbant1 | 176 | 1 |
| 65547 | 8 | 1 |
| blchaw1 | 78 | 1 |
| yercac1 | 484 | 1 |
| bobher1 | 63 | 1 |
| 41663 | 110 | 1 |
| y00678 | 221 | 1 |
| 22333 | 51 | 1 |
| bkcdon | 254 | 1 |
| anhing | 111 | 1 |

**Table 10**
Percentage distribution of files by the number of birds in Soundscape and Train Data

| # Birds per File | Soundscape (%) | Train Data (%) |
|---|---|---|
| 0 | 54.45 | – |
| 1 | 11.00 | 90.71 |
| 2 | 7.11 | 6.46 |
| 3 | 5.28 | 1.85 |
| 4 | 4.11 | 0.64 |
| 5 | 3.39 | 0.19 |

## B. Detailed Training and Inference Setup

All our models were trained using PyTorch [41] and Lightning [42] frameworks. We used the `timm` library [43] for CNN backbones.

All models, including pretraining, were trained for 50 epochs with a batch size of 64. For EfficientNetV2-S models, we used the AdamW [44] optimizer with a learning rate of $1 \times 10^{-4}$, $\epsilon = 1 \times 10^{-8}$, and $\beta = (0.9, 0.999)$. For NFNet-L0 models, RAdam [45] was used with a learning rate of $1 \times 10^{-3}$. The learning rate followed a cosine schedule down to $1 \times 10^{-6}$ without warm-up.

We used `nnAudio` [46] for on-the-fly spectrogram extraction during the forward pass. The parameters used are summarized in Table 11. After extraction, the spectrogram was converted to decibels using `AmplitudeToDb` with `top_db` = 80 and `amin` = $1 \times 10^{-10}$, then standardized and scaled to the [0, 1] range.

Regarding augmentations, MixUp was applied with a probability of 50%. Background noise augmentation was also used with a 50% probability, equally drawing noise samples from the soundscape dataset and ESC-50.

The classification head used 512 hidden channels, with a dropout of 0.25 after the CNN encoder

**Table 11**
Spectrogram extraction parameters

| Parameter | Value |
|---|---|
| Sample rate | 32,000 |
| Number of Mel bands (`n_mels`) | 128 |
| Minimum frequency (`f_min`) | 20 |
| FFT size (`n_fft`) | 2,048 |
| Hop length | 512 |
| Normalized | True |

and 0.5 after the hidden layer. A ReLU activation was used in the hidden layer. To aggregate CNN embeddings across the frequency dimension, GeM [47] pooling was employed.

Nearly all experiments were trained on random 5-second chunks from training samples (see Section 5.5). Inference on soundscapes was performed using non-overlapping 5-second windows. For validation, predictions on training files were aggregated by taking the maximum probability across segments per class.

To optimize data loading during training, audio files were converted to `h5py` [48] format, enabling byte-wise access to random 5-second chunks. For inference, model precision was reduced to FP16 and exported to the OpenVINO format [49], which also facilitates out-of-the-box deployment. On inference, spectrogram extraction was done once and reused across all ensemble models.

All training jobs were executed on two different setups:

- **Kaggle Notebook**: Used a P100 GPU. Training all 5 folds typically required 15 to 18 hours.
- **Dev Box**: Equipped with an NVIDIA GeForce RTX 4090 (24 GB). Training all 5 folds took between 5.2 and 5.7 hours, mainly due to the use of `h5py`, higher I/O throughput, and a faster CPU.

## C. Data Curation Examples

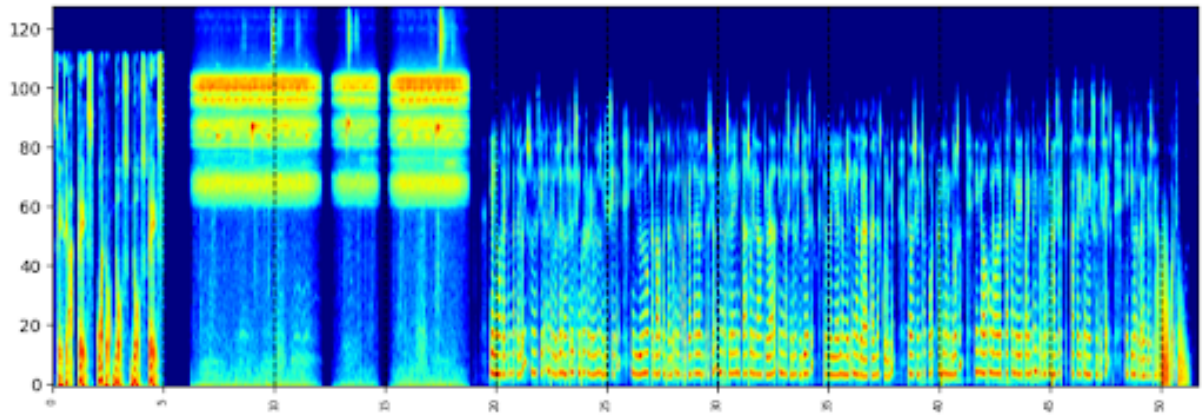


**Figure 7:** Example of alien speech occurring both before and after bird vocalization.

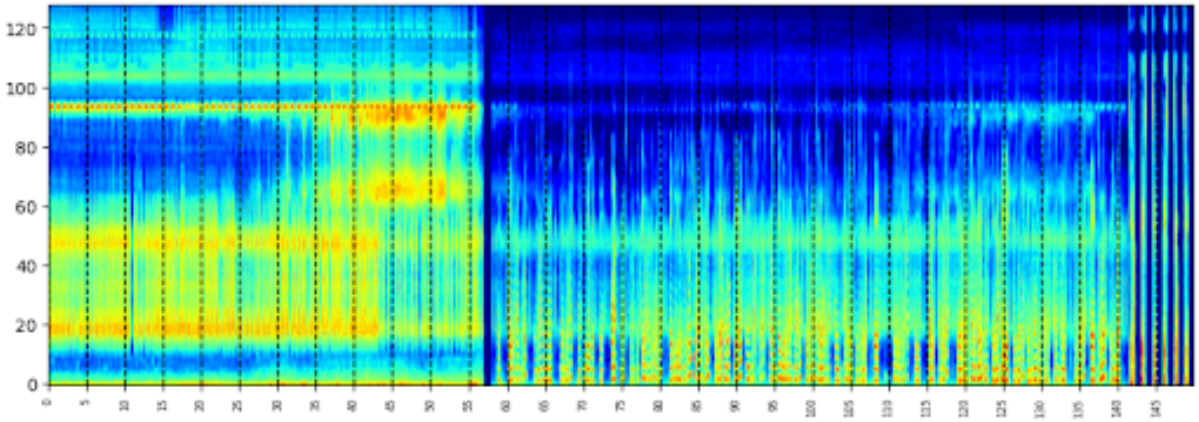


**Figure 8:** Example of overlapping speech occurring between seconds 57 and 142. The final 8 seconds are alien speech.

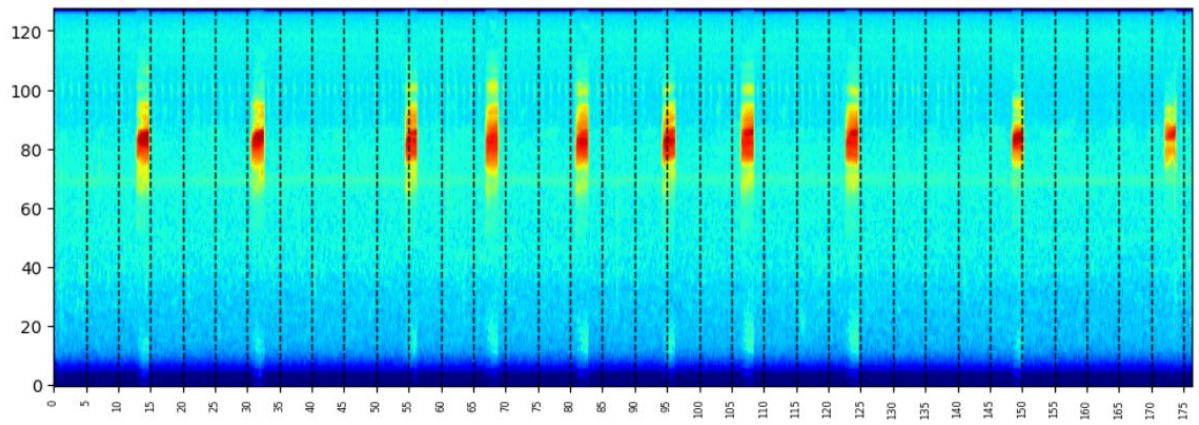


**Figure 9:** Example of a vocalization containing intermittent periods of silence.