Addressing the Challenges of Domain Shift in Bird Call Classification for BirdCLEF 2024

Emiel Witting[1], Hugo de Heer[1], Jeffrey Lim[1], Cahit Tolga Kopar[1] and Kristóf Sándor[1]

[1]TU Delft Dream Team Epoch | Dream Hall, Stevinweg 4, 2628 CN Delft, The Netherlands

Abstract

This paper presents Team Epoch IV's solution to the BirdCLEF 2024 competition, which focuses on developing machine learning models for bird call recognition. The primary challenge in this competition is the significant domain shift between the Xeno-Canto recordings used for training and the passive acoustic monitoring (PAM) soundscapes used for testing. This shift poses difficulties due to differences in recording equipment, recording conditions, and background noise, which complicates accurate species identification. We delve into the specifics of this domain shift, quantifying its impact on model performance, and we propose methods to mitigate its effects. Our approach includes a comprehensive set of data augmentations and pre- and postprocessing techniques to enhance model robustness and generalization. We performed extensive experiments to verify the effectiveness of these methods. Our findings provide a foundation for future work in addressing domain shift challenges in bioacoustic monitoring, contributing to more accurate and reliable biodiversity assessments.

Keywords

Bird Species Classification, Domain Shift, Domain Adaptation, Convolutional Neural Networks, Deep Learning, Passive Acoustic Monitoring, Kaggle Competition

1. Introduction

BirdCLEF 2024 [1] is a Kaggle competition aimed at advancing machine-learning solutions for bird call recognition, as part of LifeCLEF [2]. The primary task involves developing data processing techniques and models to identify bird species from continuous audio recordings, specifically targeting under-studied Indian bird species in the Western Ghats. This competition holds value for biodiversity monitoring, as it leverages PAM to facilitate extensive and temporally detailed surveys, contributing to conservation efforts.

Participants face several notable challenges, primarily centred around the domain shift between the training data and test soundscapes. One of the main hurdles is the difference between the Xeno-Canto recordings used for training and the PAM soundscapes used for testing. This shift is exacerbated by the fact that Xeno-Canto recordings are not expert-labelled and do not provide labels for each five-second segment, but rather for the entire file. This lack of precise labelling makes it challenging to handle secondary labels accurately. The absence of PAM data in the training set poses a significant obstacle. Participants must develop models without having access to the same type of labelled data on which their models will be evaluated, which necessitates innovative approaches to generalize effectively. Additionally, the competition imposes a strict inference time limit of two hours on a CPU, requiring efficient algorithmic implementations.

This paper presents Team Epoch IV's solution [3] to the BirdCLEF 2024 competition, with a primary focus on analyzing and addressing the domain shift challenge. We delve into the specifics of this shift and examine its impact on the discrepancy between local cross-validation scores and the public and private leaderboard scores. Our approach includes a detailed exploration of methods to mitigate these differences and enhance model performance across varied data domains.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

†These authors contributed equally.

emiel.witting@gmail.com (E. Witting); hugodeheer1234@gmail.com (H. d. Heer); Jeffrey-Lim@outlook.com (J. Lim); cahittolgakopar@gmail.com (C. T. Kopar); Emherk512@gmail.com (K. Sándor)

https://github.com/EWitting (E. Witting); https://github.com/hjdeheer (H. d. Heer); https://github.com/Jeffrey-Lim (J. Lim); https://github.com/tolgakopar (C. T. Kopar); https://github.com/emherk (K. Sándor)

The paper is structured as follows: Section 2 describes our implementation strategy, including environmental setup, data preprocessing, data augmentation, model selection, and postprocessing techniques. Section 3 discusses the domain shift between training and test data. Section 4 presents our experiments and results, including an ablation study and seed stability analysis. Section 5 discusses our findings, and Section 6 concludes with future work and acknowledgements.

2. Implementation

In this section, we detail our implementation strategy employed for our participation in the BirdCLEF 2024 competition. Our approach encompasses environmental and training setup, data preprocessing, data augmentation, model selection, and postprocessing techniques.

2.1. Environmental setup

During the competition, we collaborated as a team. Instead of working in notebooks, which does not allow for streamlined collaboration, we developed and used our machine learning framework Epochalyst [4]. This package contains many modules and classes extracted from previous Epoch [5] competition experience to start new competitions quickly. Epochalyst makes use of hydra to load in configuration `.yaml` files that specify full training or ensemble runs and instantiate elements directly into Python objects for efficient development. We used Rye [6] for project & package management and designed a custom lazy loading multiprocessing pipeline for loading audio using Dask [7] and Librosa [8]. PyTorch [9] was utilized as the main framework for training, with additional libraries such as Timm [10] for using various 2D Convolutional Neural Network architectures. Additionally, for an extra ~2× inference speed up, ONNX [11] and OpenVINO [12] were used to maximise performance. Models were trained on on-site hardware running Linux [13], specifically on PCs running AMD Ryzen 9 7950X 16-Core Processor (96GB RAM) with an NVIDIA RTX A5000 GPU using Python 3.10.13. Model training and run artefacts were logged on Weights & Biases [14] to keep a clear overview of all of our experiments.

2.2. Data Preprocessing

The BirdCLEF 2024 training dataset consists of 24459 audio `.ogg` files uploaded by users of Xeno-Canto [15], consisting of 182 different bird species. All the training audio has been resampled to 32 kHz to match the test soundscape sampling rate. We did not obtain improved results pretraining on more data from previous BirdCLEF competitions, therefore we only used this year's data for our final submission. For training, we used a 5-fold CV with a stratified split based on the primary label of the audio file. This ensures that the species are equally represented in each fold. Taking the first 5 seconds of each audio file seemed to be optimal since the bird calls of the recordings had a higher probability of appearing early in the uploaded recordings. Some Xeno-Canto files also contained secondary labels of bird species that appeared in addition to the primary bird. For these, we set the secondary labels to 0.5 and the primary labels to 1, because the primary birds were consistently more audible in the audio files compared to the secondary birds.

2.3. Data Augmentation

We have implemented several data augmentation techniques to increase the robustness of our models and to address the domain shift between the training data and the test soundscapes. Our full augmentation pipeline can be seen below. Some of the augmentations are 1D, which means they are applied to the raw audio signal. Afterwards, we converted the signal to Mel spectrograms of 256×256 pixels, with a frequency range of 1Hz to 16kHz, which are then further normalized so that all values are in the range of 0 to 1. As a last step in our custom dataset, some 2D augmentations are applied.

1D-Augmentations

- Randomly shifting the phase of each frequency component of the signal with $p = 0.5$ and a shift_limit $= 0.5$.[1][2]
- Randomly shifting the amplitude of each frequency component of the signal with $p = 0.5$.
- MixUp [16] with $p = 0.5$:
  Linearly interpolating both features and labels of two samples, with random weights.
- CutMix [17] with $p = 0.5$:
  Randomly cropping and replacing part of a sample with another sample. The labels are averaged linearly with weights proportional to the length/area of each sample.

2D-Augmentations

- CutMix with $p = 0.5$.

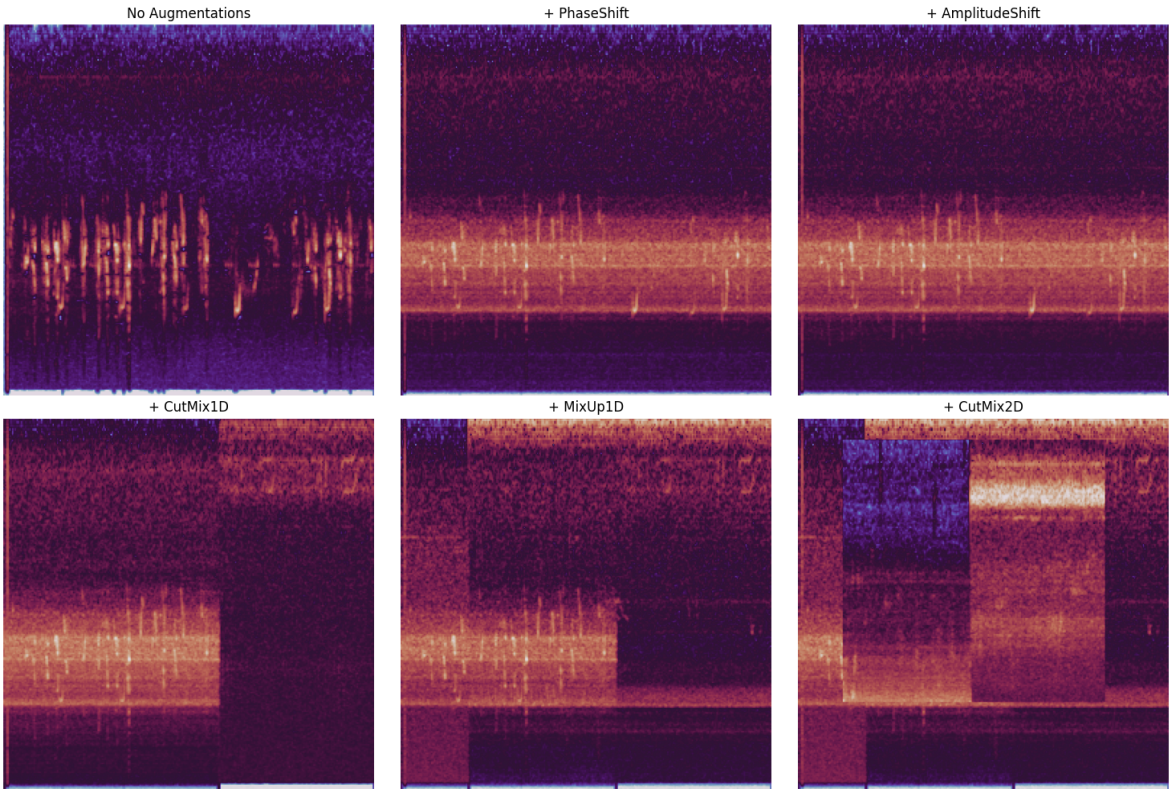


Figure 1: Example of our augmentation pipeline applied to a training image. (Left to right, top to bottom)

Figure 1 above visualizes our augmentation pipeline. The phase shift aims to simulate background noise in bird regions, in an effort to reduce the shift between the clear training examples and the noisy soundscapes. Amplitude shift amplifies different frequencies in the signal domain to enhance robustness against bird volume variance, since there are birds in the soundscapes that are in close proximity to the recording location whilst others might be located further away. Afterwards, the CutMix1D and MixUp1D are applied in the signal domain to improve learning when there are multiple birds in the same audio file, which is a common occurrence for the soundscapes. Finally, a CutMix2D is applied after converting the previous pipeline to a Mel spectrogram. An ablation study of these augmentations can be found in section 4.1.

[1] This does not influence the magnitude spectrum taken over the whole recording, but when windowed magnitude spectra are extracted it has the effect seen in Figure 1

[2] shift_limit in the range [0,1] corresponds to a phase shift of $[0,2\pi]$

2.4. Models

We mainly used Timm [10] for straightforward model development where we experimented with various architectures. Some of the best encoders that we have found were: `convnext_tiny` and `eca_nfnet_l0`. The `convnext_tiny` model got the highest public leaderboard score of 0.701 while we observed it being more unstable over multiple submitted training runs. `eca_nfnet_l0`, on the other hand, had a slightly lower public score of 0.688 but we found it to be a more stable model during experimentation. We decided to submit these two models: the more stable one and the less stable one but with a higher public score. During training for 50 epochs total with an initial learning rate of $1e{-}4$, Binary Cross-Entropy [18] loss was used with an AdamW [19] optimizer. A single cycle `CosineAnnealing` learning rate scheduler was employed with a slight warmup of 2 epochs to ensure initial stability. Furthermore, models had a sigmoid activation function to ensure that the logits ranged between 0 and 1. Local evaluation was done on every 5 seconds of each file, using the AUROC [20] metric where we observed a significant shift between our local scores and public scores. We were able to optimize our local scores to ~0.995 AUROC, by adding dropout, training on multiple datasets from previous BirdCLEF [3] competitions and including additional augmentations. However, any optimization above ~0.98 local, caused the public score to drop significantly. Therefore observing an overfitting pattern on Xeno-Canto data while reducing the performance on the soundscapes, proposing us to focus on minimizing the shift and not optimizing on training data.

2.5. Postprocessing

The test soundscapes are 4 minutes long, where we have to predict for each 5-second window resulting in 48 predictions per soundscape. We calculate the mean bird species probability per soundscape over the 48 windows and multiply each individual prediction by the mean of the soundscape it is in. The reasoning behind this was that we saw that birds usually appear multiple times per recording, so the mean should be high for birds that are truly in the audio. This improved our scores consistently with ~0.02 on public and private.

3. Domain shift

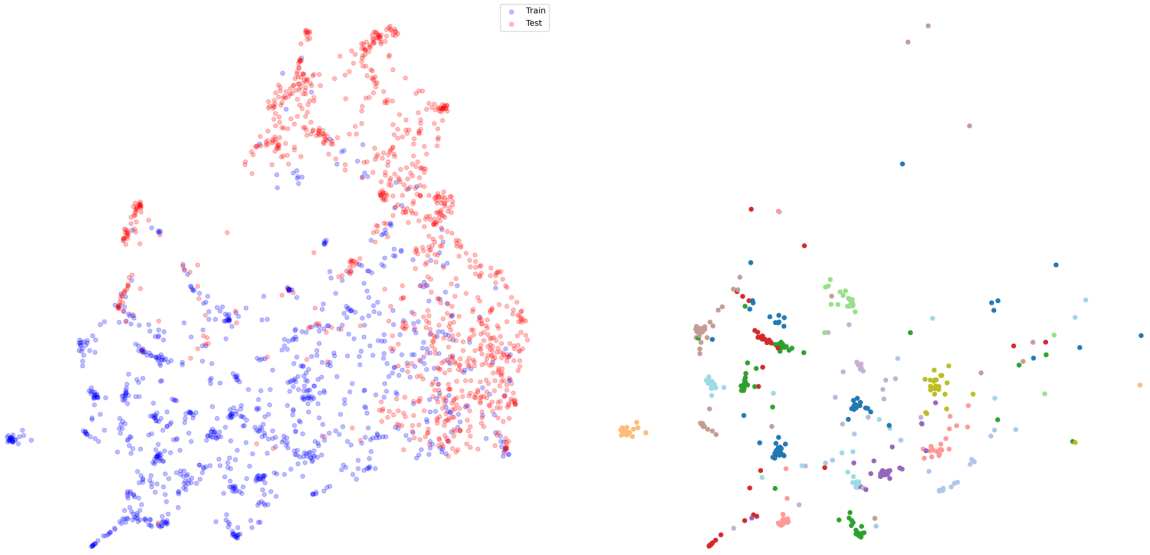
The training data for this competition was sourced from a different domain than the test set for which the models are intended. This poses a problem that is highly relevant in non-competition settings, where (labelled) test data is not always available. The impact quickly becomes obvious, when observing that models can achieve above 0.99 AUROC on held-out training data, but score below 0.70 AUROC on the test set. In this section, we will hypothesize components of the discrepancy, guided by statistical analysis, knowledge about the data source, and visual inspection. Furthermore, we attempt to quantify the domain shift and measure the impact of techniques to mitigate the shift.

3.1. Mapping the datasets

We explored the train dataset and looked for differences with the unlabeled soundscapes, by making a visual overview. To ensure that we organize the audio in the way our model perceives it, we compared the activations of the last hidden layer of a baseline model, instead of the raw input. For both domains, the first five seconds of one thousand unique recordings were fed through the model. The activations were then projected using UMAP[21] onto $\mathcal{R}^2$. This shows that there is partial overlap between the domains, and part of the test domain seems completely outside of the training distribution (Figure 2a).

We then plotted the corresponding spectrograms at the positions of their UMAP embedding. This allowed us to understand and identify the different regions of the dataset manually, as shown in Figure 3. Quadrants I and II contain mostly no-calls. It makes sense for these to be outside of the training distribution, which should only have labelled bird calls. The difference appears to be that II is fully

[3] BirdCLEF 2020, 2021,2022 and 2023 Xeno-Canto and labelled soundscape data retrieved from Zenodo.



(a) Distribution shift (b) Top-20 known labels

Figure 2: UMAP Projection of activations of the last hidden layer, for 1000 train audio samples, and 1000 unlabeled soundscapes. Uses only the first five seconds per recording. (a) Shows all samples, coloured blue for train and red for test data. (b) Training samples for the 20 most common classes, coloured by class. This confirms that the embedding can achieve class separation for train audio.

quiet, or at least has uniform noise, whereas I has noisy recordings with sounds other than birds. Most bird calls appear in III and IV. III Contains mostly training data, which seems to be characterized by low-noise, high-contrast bird call recordings. Towards region IV there is a gradient of increasing amounts of test data. This seems characterized by high background-noise images with less contrast (the background looks consistently brighter) and horizontal stripes. We assume the horizontal stripes are likely insects, such as cicadas. Note that there are in fact some training samples that fit into this distribution, as can be seen in both Figure 2a and in the bottom right of 3.

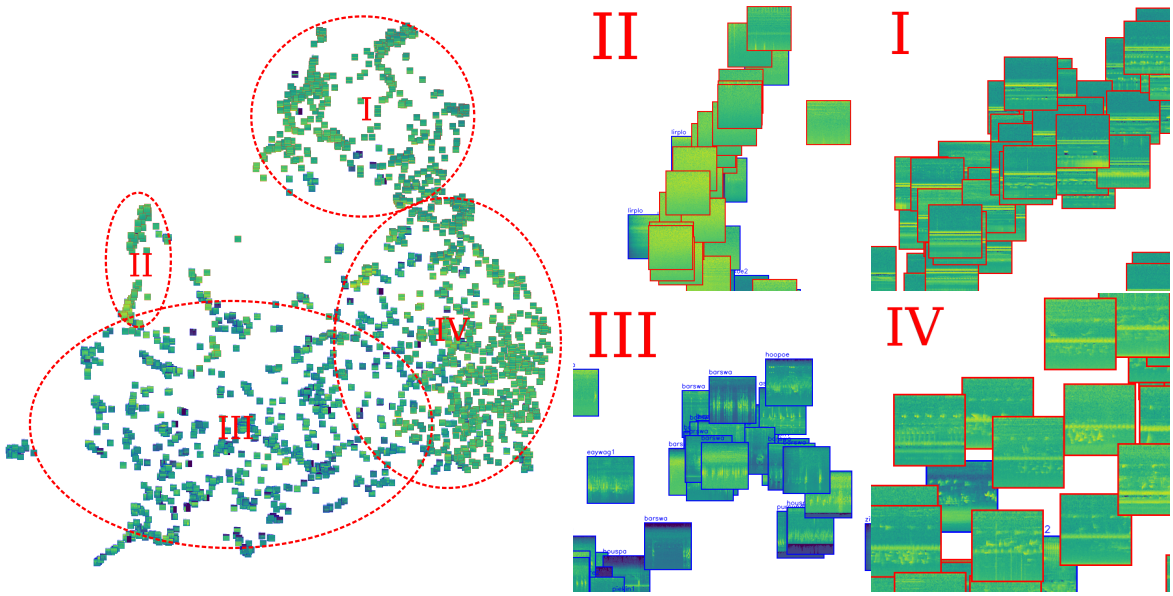


Figure 3: The same UMAP projection as in Figure 2, where the samples are shown using their spectrograms. Blue outlines indicate train audio, and red outlines indicate unlabeled soundscapes. Training audio is annotated with the label. (left) The four quadrants we can divide the data into. (right) Zoomed-in sections of each quadrant.

3.2. Shift causes

It is important to be cautious about assuming the nature of the problem. High train performance and low test performance on another domain seem to warrant (unsupervised) domain adaptation, to solve the apparent problem of domain shift. Well-documented forms include covariate shift, prior shift, or concept shift [22]. These might be approached with feature-based sample weighting, class-based sample weighting, and deep domain adaptation methods respectively. However, all of these forms rely on the assumption that there is only one type of shift at a time and that some other factor remains constant. Furthermore, it is possible that generalisation is not an issue, and that the drop in score can be explained solely by the fact that the test domain is just uniformly more difficult and ambiguous.

With this caution in mind, we hypothesized three main contributors for the drop in score:

1. The models underperform on no-call audio, which the Xeno-Canto training data does not contain.
2. The PAM test data are inherently more difficult and ambiguous to classify, even for models trained on it.
3. The PAM test data is shifted into feature distributions that our model has not encountered or generalized to properly during training.

We exclude prior shift, or label balance, as a root cause. This is because the scoring metric is mostly class-balance invariant.

Hypothesis 1 was confirmed by measuring the predictions on test samples from regions I and II that we confirmed to be no-calls. We observed that our model was consistently making predictions at around $0.5 \sim 0.6$ confidence. Those false positives occur across a handful of species. Mostly `browow1`, `comior1` and `comkin1` for region I, and `woosan` for region II. We estimate this partly being due to a random bias, and correlated background noise. We have seen false `browow1` positives for several models, possibly due to those samples being recorded for a nocturnal species with mostly quiet recordings and sometimes insects.

Hypothesis 2 allows the possibility that test-like data is in fact represented in the train data, but at lower proportions. We might approximate the difficulty for the PAM-like samples, by measuring train scores in region IV. This resulted in a class-mean AUROC of 0.985. This is clearly significantly higher than the leaderboard test score, also when regarding the small sample size (154 samples with 75 unique species). This evidence contradicts hypothesis 2. A reason for not rejecting it fully is that those train samples might not be representative of test data, and that they differ along a dimension that is not captured by UMAP.

Hypothesis 3 is the standard problem of models not generalizing to data that is outside of the training distribution. The main differences we noticed visually were the decreased contrast (low signal-to-noise ratio) and horizontal stripes from possibly insects. Furthermore, we observed more overlapping bird calls in the test soundscapes than in train audio. A participant in BirdCLEF 2023 mentioned reverb [23]. This might also be the case, although we have not had the opportunity to verify this or test reverb augmentations.

3.3. Shift mitigation

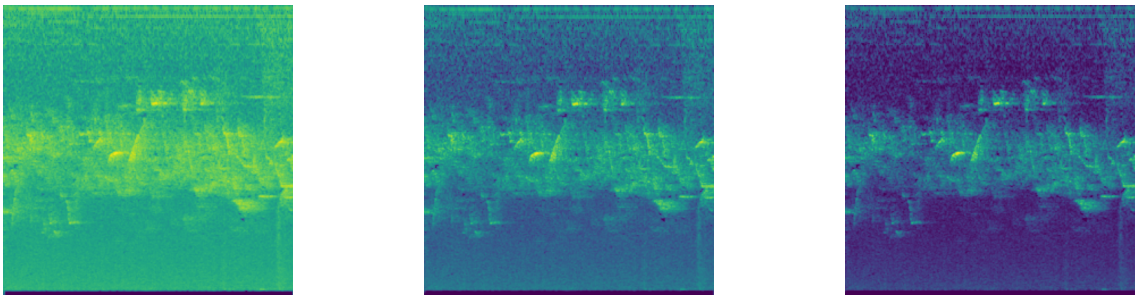
3.3.1. Call or no call classification

To mitigate the fact that our model underperforms on no-call audio, a two-stage pipeline was introduced. The first stage consists of training a model on the `Freefield1010` [24] dataset to perform a binary classification task for every 5-second window to predict if there is a call or no call from birds, a f1-score of 0.810 was achieved for stage one. If the first stage predicted that there was a no call in a 5-second window of a soundscape, all predictions were set to 0 and the second stage for this window was skipped, also saving important inference time. After empirical visual inspection of the soundscapes, the two-stage appear to be correct for silent soundscapes with an example in Figure 9 and 10 illustrating predictions of our best submission compared to our two-stage approach on a silent soundscape. Interestingly, against our expectations, our public scores did not improve by submitting our two-stage approach. Further

investigation with the labels of the test soundscapes is recommended to detect where our two-stage model is making its mistakes.

3.3.2. Test Audio Scaling

In order to remove the shift between distributions as mentioned in hypothesis three, we tried two techniques. The first is to scale down the test audio during inference. Because we min-max scaled our spectrograms, this is analogous to increasing ϵ in the logarithmic scaling $\log(x + \epsilon)$ that we applied to spectrograms. We scale the audio down by a factor of $1/100$, which we found through empirical experimentation. The effect is an increased contrast, which visually makes the test data look more similar to the train data. We consistently achieved higher scores for both the public and private leaderboards as a result.



(a) Factor 1 (b) Factor 1/100 (c) Factor 1/500

Figure 4: Scaling Effect on the first 5 seconds from `unlabeled_soundscapes/1527167.ogg`

3.3.3. Frequency-based noise removal

The second technique aims to remove ambient noise. It treats the audio as a sum of infrequent (bird) noises, and background noise that is stronger in some frequencies than others, but is constant over time. Visually, this means removing all horizontal stripes from spectrograms.

To obtain a robust estimate of the background noise level per frequency, the quantile $q = 0.25$ was used per row of the spectrogram. This implictly assumes that a bird call does not occupy the same frequency for more than three-quarters of the sample. If that assumption is true, the value will not be impacted by outliers from bird calls, which the mean would be sensitive to. This is then subtracted from the original image, an example is shown in figure 5.

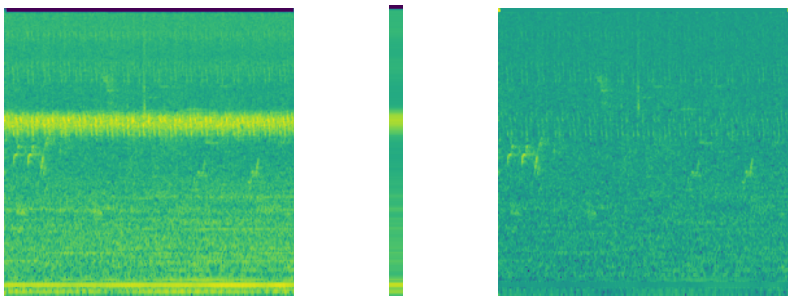


Figure 5: (left) Original sample. (middle) The q=0.25 quantile per frequency, one dimensional. (right) after subtracting the quantile.

3.3.4. Domain distance

To quantify to which extent the two domains were becoming more similar, we used a modified Frechet Inception Distance [25]. FID compares distributions of the activations of the last hidden layer of an

Inception-v3 network between two datasets. We use the activations of our best-selected submission model instead. Because the goal was not to remove the discrepancy between call and no-call, that separation needed to be preserved, only regions III and I were used. We hypothesised that this could be used to estimate the impact of a shift mitigation technique, before training a model and evaluating it with test labels.

3.3.5. Deep Domain Adaptation

Some methods are developed specifically to mitigate the domain shift while training a deep learning model. Examples are DANN [26] and MDD [27]. They take labelled train data and unlabeled test samples. We did not have success with these techniques, however. In part, this was due to the difficulty of tuning hyperparameters for the adversarial networks that they contain, which are prone to instability. An argument for why these techniques might not be suitable at all without modification is that their objective includes removing as much differences between train and test as possible. This could cause issues when a major difference is the existence of no-calls only in test. This might mean that optimising the objective requires removing the ability to tell bird calls from silence.

4. Experiments

This section contains experiments regarding our ablation study in section 4.1 and seed stability experiments in section 4.2. To verify the significance of improvements in the ablation study, we also investigated the effect of randomness on the leaderboard performance. For this, experiments with the same configuration but with various seeds that affect the data split, augmentations, and model weight initialization the stability of our models have been trained and evaluated.

4.1. Ablation Study

In our ablation study, we aim to analyse the effect of our augmentations. We performed 6 training runs of our best submission model with 5-fold CV, where we added one augmentation at a time. We submitted each fold individually, resulting in 30 total submissions. From Figure 6 we can observe a very high score variance within folds and a positive correlation of 0.85 between public and private. The *Leaderboard* group in the boxplot contains a weighted average of the public and private score and is calculated as follows: 35% Public Score + 65% Private Score. MixUp1D resulted in the most significant improvement of ~0.03 on the Kaggle leaderboard.

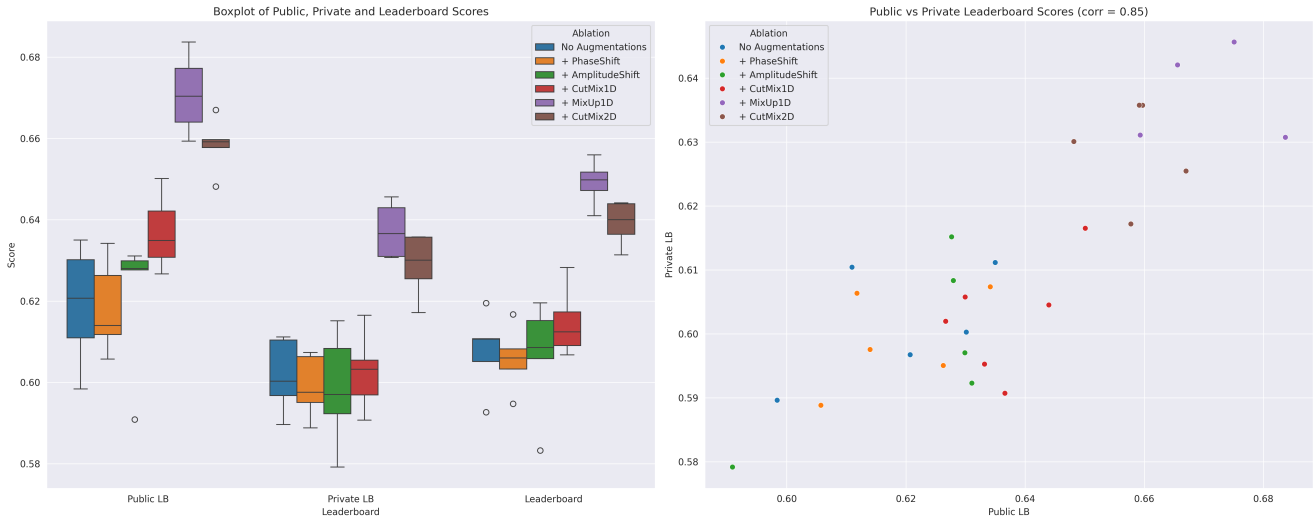
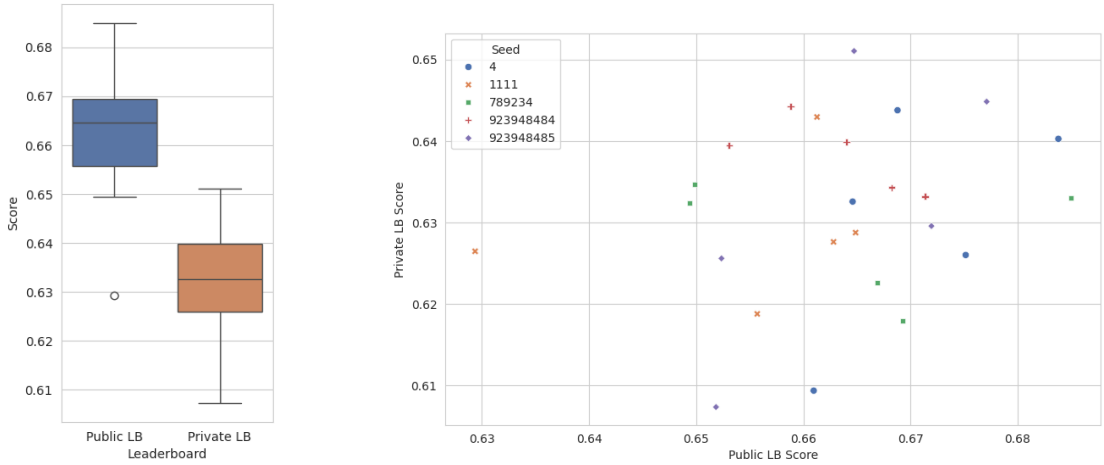


Figure 6: Ablation study of augmentations (left) and Public and Private correlation (right)

4.2. Seed Stability

For our seed stability experiment, we used the model pipeline from our best submission and retrained that model 5 times with different seeds for 5-fold CV each. We made late submissions on Kaggle for each fold, resulting in a total of 25 variations of the same model. The public and private leaderboard scores of these models are shown in figures 7a and 7b.



(a) Distribution of Public & Private LB Scores (b) Correlation between Public & Private LB Scores: 0.25

Figure 7: Seed Stability: Public and Private Leaderboard Score

There is a significant variance for both the public and private leaderboard scores with the same model. More specifically, the public leaderboard scores have a standard deviation of 0.01197 and the private leaderboard scores have a standard deviation of 0.01091. Furthermore, we can observe a slight correlation of 0.25 between public and private leaderboard scores by only varying the seed of the run configuration.

It is worth noting that while various assembling techniques can be used to stabilize models, this is not always possible due to the strict CPU inference time limit.

4.3. Shift mitigation

To evaluate the effect of shift mitigation using the test-time scaling and frequency-based filter, 5 submissions were made for each, along with a baseline. For the frequency normalized the model was trained again with the filter applied, but the test scaling was performed at inference time with the original model weights. Again, these 5 submissions are the results of training on a different fold. The results are shown in 8, the score is calculated as 35% Public Score + 65% Private Score. Both methods improve the baseline, audio scaling most significantly.

For these three techniques, we measure the distance between train and test distributions as described in section 3.3.4. This was measured only for data that was in regions III and IV in the original UMAP projection, with the same model that generated figure 2. For the baseline, the modified FID was 41.4, by applying the frequency-based filter to all data, the distance shrank to 37.6. When instead rescaling only the test audio by $1/100$, it increased to 46.7.

5. Discussion

Augmentations

From our ablation study in section 4.1 we found that MixUp1D was one of the best-performing augmentations. We suppose this is due to the soundscapes consisting of more birds simultaneously compared

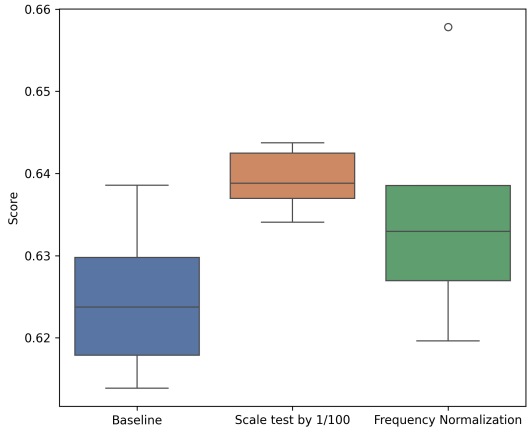


Figure 8: Combined leaderboard scores across five folds of the same model. Comparing the baseline against test-time audio rescaling and the frequency-based filter.

to our training data. MixUp increases the models' capability of learning different birds at the same time. The CutMix augmentations did not result in a significant improvement, after further analysis we observed that CutMix often cuts out a bird call and replaces it with a silent section of another bird audio fragment, therefore confusing the model with learning it to annotate a silence with a bird call. We consider that the PhaseShift augmentation was set to be too extreme, therefore adding too much noise and reducing the models' capacity to learn the visual bird call patterns. On the contrary, the AmplitudeShift performance was inconclusive and we suggest tuning it to higher intensity to have more effect.

Seed stability

An interesting observation was that we found that our models were in general quite unstable. During our experimentation we did not find any way of locally evaluating our models, therefore we relied on the public leaderboard. In our experiments specified in section 4.2, we perceived that the same model configuration trained on a different seed could significantly change the public and private leaderboard performance. Therefore, indicating that randomness was highly involved during our experimentation phase. During this phase, we implemented novel ideas and the only way we found to evaluate the idea was by making a submission. However, we might have discarded ideas that got an 'unlucky' low public score due to the elevated randomness, which was better if we had analysed the average over multiple submissions. Furthermore, it is also interesting to note that the public and private scores have a slight correlation on submitting different seeds, which could indicate that optimizing the seed on the public leaderboard could also transfer to a higher private score.

Shift mitigation

Visualizing and interacting with the datasets helped us understand the differences between train and test. This guided us towards two techniques that visually removed the shift and also improved scores on the test set. However, we are not certain about the impact of the no-call segments. While it seemed that there were many false positives, the two-stage approach did not solve this problem. This might be caused by not optimizing the two-stage model sufficiently, but it is also surprising that out of the top 5 solutions for BirdCLEF 2024, nobody seemed to get consistent improvements by using a call/no-call model, even though it has been used successfully in other editions. Furthermore, it is not clear what proportion of the score drop is caused by the shift in regions III and IV as in figure 3, or false positives. We are not able to confirm this without access to the labels.

Unfortunately, decreasing the FID distance between domains does not guarantee an improved test score and vice-versa. While the distance only increased for scaling and decreased for the filter, the score improved for both. The distance change might be explained by the fact that scaling was only applied to the test data, which introduces a synthetic shift that can be measured but does not negatively impact the model. The frequency-based noise filter was applied on both datasets instead of only one and removed shift distance as intended.

6. Conclusion

In this paper, we presented our solution for the BirdCLEF 2024 competition, focusing on the challenge of domain shift between the training and test datasets. Our approach includes shift mitigation through data augmentation and preprocessing. We evaluated the stochasticity of the results and performed experiments with thorough 5-fold validation.

We found that:

- Domain Shift Mitigation: Applying frequency-based noise removal and scaling test samples, as guided by exploratory data analysis, proved successful. This introduces a novel filter that could be applicable to other PAM audio classification problems and generally highlights the importance of investigating the aspects of domain shift.
- Data Augmentation: MixUp1D applied to audio is a particularly effective technique. This is likely due to the presence of multiple bird calls per recording in the test soundscapes. Other augmentations such as CutMix and PhaseShift did not yield improvements, these might need to be adapted or excluded in similar experiments.
- Seed stability: Our seed stability experiments revealed substantial variance in public and private leaderboard scores, emphasizing the impact of randomness in model training. This underscores the importance of averaging results over multiple seeds to obtain a reliable performance estimate. It also implies that conclusions based on the competition outcome should be drawn with caution if scores are close.
- Call/No Call Classification: Implementing a two-stage pipeline for call/no call classification did not result in the expected score improvements. This suggests that our current implementation may need refinement or that the issue of false positives might be more complex than anticipated.

Overall, our findings indicate that addressing domain shift is crucial for achieving robust performance in bird call classification tasks. Our methods provide a foundation for future work, including further refinement of data augmentation techniques, deeper analysis of domain shift, and more sophisticated model evaluation strategies.

In conclusion, while we achieved notable improvements, the BirdCLEF 2024 competition highlighted the ongoing challenges in developing models that generalize well across different acoustic environments. Our results underscore the need for continuous innovation and experimentation in tackling domain shift and enhancing model robustness.

6.1. Future work

Looking ahead, several avenues for future research and experimentation emerge from our findings. First of all, during our experimentation phase, we might have discarded ideas because they were based on a single submission. Due to the observed impact of randomness on the scores, several ideas that were initially discarded are worth investigating again, including:

- Alternating Ensemble: For every 4-minute soundscape, corresponding to 48 windows, let n different models predict the windows alternately, followed by averaging neighbouring window predictions. In this way, we are able to ensemble without increasing inference time.
- Pretraining on previous years: Rerun experiments where we append BirdCLEF data including soundscape PAM data from Zenodo.

- Two-stage refinement: Refining the two-stage pipeline approach and incorporating more sophisticated methods for distinguishing between bird calls and background noise in addition to the freefield1010 dataset.
- Longer Window Models: Experimenting with models that process longer audio windows (e.g., 10 seconds) could provide more context and improve classification accuracy.
- Multi-Channel Spectrograms: Investigating the use of multi-channel spectrograms to capture richer audio information.

Secondly, obtaining and analyzing the labels of the test soundscapes would allow us to validate our hypotheses about domain shift and the effectiveness of our mitigation techniques. Finally, we encourage the competition hosts to analyze the best solutions to the competition again. It could be very insightful to measure the score when excluding no-calls, or excluding overlapping bird calls, to isolate the effects.

Acknowledgments

We would like to thank the organizers of the BirdCLEF 2024 competition and all involved institutions. We extend our thanks to all the participants of the BirdCLEF 2024 competition who were active in the Kaggle discussion forums for their ongoing efforts in advancing the field of bioacoustics and biodiversity monitoring. Your dedication and collaboration are instrumental in driving forward conservation efforts worldwide.

References

[1] S. Kahl, T. Denton, H. Klinck, V. Ramesh, V. Joshi, M. Srivathsa, A. Anand, C. Arvind, H. CP, S. Sawant, V. V. Robin, 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, Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024).

[2] A. Joly, L. Picek, S. Kahl, H. Goëau, V. Espitalier, C. Botella, B. Deneu, D. Marcos, J. Estopinan, C. Leblanc, T. Larcher, M. Šulc, M. Hrúz, M. Servajean, et al., Overview of lifeclef 2024: Challenges on species distribution prediction and identification, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2024.

[3] J. Lim, E. Witting, H. de Heer, C. T. Kopar, K. Sándor, Identifying Bird Calls, 2024. URL: https://github.com/TeamEpochGithub/iv-q4-birdclef-2024.

[4] J. Lim, E. Witting, H. de Heer, C. T. Kopar, J. van Selm, A. Ebersberger, G. Dumont, K. Sándor, D. De Dios Allegue, Epochalyst, 2024. URL: https://github.com/TeamEpochGithub/epochalyst.

[5] J. Lim, E. Witting, H. de Heer, C. T. Kopar, J. van Selm, A. Ebersberger, G. Dumont, K. Sándor, D. De Dios Allegue, 2024. URL: https://teamepoch.ai/.

[6] A. Ronacher, Rye: a Hassle-Free Python Experience, 2024. URL: https://rye.astral.sh/.

[7] Dask core developers, Dask | Scale the Python tools you love, 2024. URL: https://www.dask.org/.

[8] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. van Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, J. Mason, K. Lee, B. Steers, E. Halvachs, C. Thomé, F. Robert-Stöter, R. Bittner, Z. Wei, A. Weiss, E. Battenberg, K. Choi, R. Yamamoto, C. Carr, A. Metsai, S. Sullivan, P. Friesch, A. Krishnakumar, S. Hidaka, S. Kowalik, F. Keller, D. Mazur, A. Chabot-Leclerc, C. Hawthorne, C. Ramaprasad, M. Keum, J. Gomez, W. Monroe, V. A. Morozov, K. Eliasi, nullmightybofo, P. Biberstein, N. D. Sergin, R. Hennequin, R. Naktinis, beantowel, T. Kim, J. P. Åsen, J. Lim, A. Malins, D. Hereñú, S. van der Struijk, L. Nickel, J. Wu, Z. Wang, T. Gates, M. Vollrath, A. Sarroff, Xiao-Ming, A. Porter, S. Kranzler, Voodoohop, M. D. Gangi, H. Jinoz, C. Guerrero, A. Mazhar, toddrme2178, Z. Baratz, A. Kostin, X. Zhuang, C. T. Lo, P. Campr, E. Semeniuc, M. Biswal, S. Moura, P. Brossier, H. Lee, W. Pimenta, librosa/librosa: 0.10.2.post1, 2024. URL: https://doi.org/10.5281/zenodo.11192913. doi:10.5281/zenodo.11192913.

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[10] R. Wightman, Pytorch image models, https://github.com/rwightman/pytorch-image-models, 2019. doi:10.5281/zenodo.4414861.

[11] O. R. developers, Onnx runtime, https://onnxruntime.ai/, 2021. Version: 1.17.3.

[12] I. Corporation, Openvino, https://docs.openvino.ai/, 2018. Open-source toolkit for optimizing and deploying deep learning models.

[13] Canonical Ltd., Ubuntu 23.10 (mantic minotaur), https://releases.ubuntu.com/23.10/, 2023. Operating system release.

[14] L. Biewald, Experiment tracking with weights and biases, https://www.wandb.com/, 2020. Software available from wandb.com.

[15] X. canto Foundation, Xeno canto, https://xeno-canto.org/, 2005.

[16] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[17] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, CoRR abs/1905.04899 (2019). URL: http://arxiv.org/abs/1905.04899. arXiv:1905.04899.

[18] P. Shukla, How did binary cross-entropy loss come into existence?, Towards AI (2023).

[19] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[20] T. Fawcett, An introduction to roc analysis, Pattern recognition letters 27 (2006) 861–874.

[21] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, arXiv preprint arXiv:1802.03426 (2018).

[22] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognition 45 (2012) 521–530. URL: https://www.sciencedirect.com/science/article/pii/S0031320311002901. doi:https://doi.org/10.1016/j.patcog.2011.06.019.

[23] M. Lasseck, Bird species recognition using convolutional neural networks with attention on frequency bands, CEUR Workshop Proceedings (2023). URL: https://www.CEUR-WS.org/vol-3497/paper-175.pdf. doi:https://doi.org/10.1016/j.patcog.2011.06.019.

[24] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, M. D. Plumbley, An open dataset for research on audio field recording archives: freefield1010, arXiv:1309.5275, 2013. doi:10.48550/arXiv.1309.5275.

[25] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. arXiv:1706.08500.

[26] A. Sicilia, X. Zhao, S. J. Hwang, Domain adversarial neural networks for domain generalization: When it works and how to improve, 2022. arXiv:2102.03924.

[27] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, H. T. Shen, Maximum density divergence for domain adaptation, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (2021) 3918–3930. URL: http://dx.doi.org/10.1109/TPAMI.2020.2991050. doi:10.1109/tpami.2020.2991050.

A. Extra visualizations

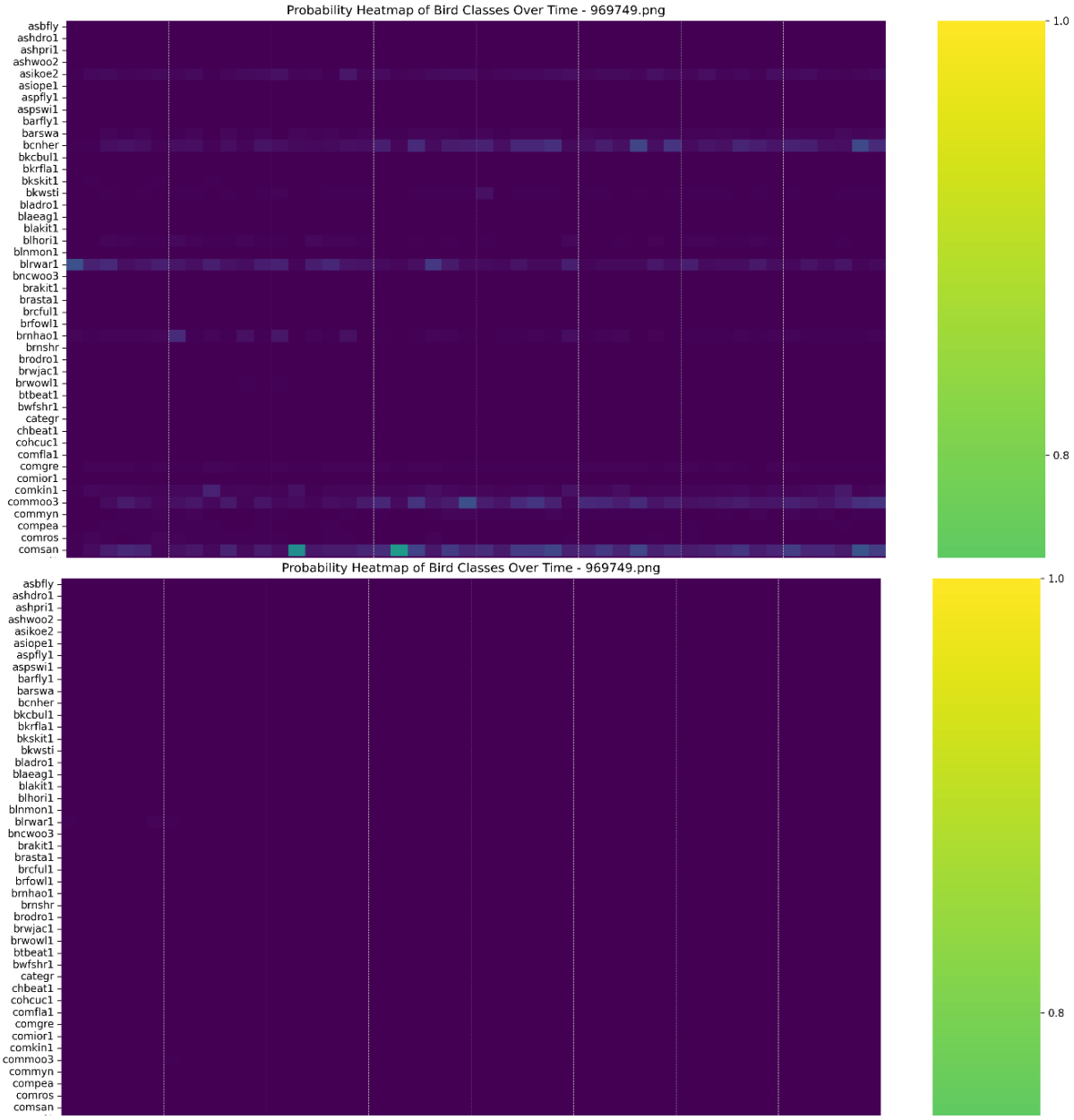


Figure 9: Predictions on `unlabelled_soundscapes/969749.ogg`. Above is our best submission, below is our no-call two-stage config trained on `freefield1010`.

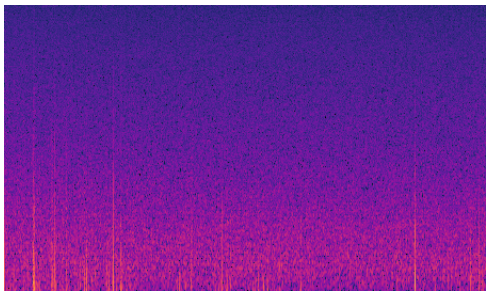


Figure 10: Melspectrogram of `unlabelled_soundscapes/969749.ogg`.

B. Results

This section contains additional raw results from our experiments.

B.1. Seed Stability

Table 1
Public and private scores of our seed stability experiments referring to section 4.2

Seed	Fold	Local	Public LB	Private LB
4	0	0.9725029805290062	0.664600	0.632564
	1	0.9721286820877476	0.668787	0.643774
	2	0.977704551531352	0.683795	0.640266
	3	0.9681365574407902	0.660968	0.609359
	4	0.9710765974889252	0.675150	0.626007
1111	0	0.9696436449770318	0.664878	0.628754
	1	0.9769452863312144	0.662820	0.627614
	2	0.9751090021447962	0.629374	0.626460
	3	0.971353063769448	0.661280	0.642945
	4	0.97246352665961	0.655700	0.618778
789234	0	0.9711130756503932	0.684964	0.633027
	1	0.9779123620481512	0.666920	0.622572
	2	0.9718111096006832	0.649368	0.632342
	3	0.9753510191654698	0.649850	0.634721
	4	0.974674509187293	0.669299	0.617909
923948484	0	0.972626292971828	0.671356	0.633187
	1	0.9726160377881716	0.658814	0.644254
	2	0.9756232466898276	0.664013	0.639876
	3	0.971870530276624	0.653071	0.639465
	4	0.9724243997642767	0.668238	0.634289
923948485	4	0.9750817450565112	0.677104	0.644832
	1	0.9735544201342272	0.652362	0.625589
	2	0.9752181693908796	0.671951	0.629559
	3	0.9721216597194158	0.664733	0.651040
	4	0.9679696078489052	0.651850	0.607342

B.2. Ablation study

Table 2
Public and private scores of the augmentation ablation study performed in section 4.1.

Ablation	Fold	Public LB	Private LB
No Augmentations	0	0.635002	0.611164
	1	0.610959	0.610434
	2	0.630144	0.600282
	3	0.620723	0.596746
	4	0.598396	0.589628
+ PhaseShift	0	0.626281	0.595057
	1	0.605735	0.588839
	2	0.613994	0.597562
	3	0.611792	0.606357
	4	0.634171	0.607367
+ AmplitudeShift	0	0.629907	0.597048
	1	0.590896	0.57918
	2	0.627677	0.61518
	3	0.631077	0.592303
	4	0.627975	0.608342
+ CutMix1D	0	0.650119	0.616519
	1	0.626682	0.601974
	2	0.633204	0.595274
	3	0.636626	0.590721
	4	0.629955	0.605772
+ MixUp1D	0	0.644002	0.604525
	1	0.659341	0.631091
	2	0.665605	0.642062
	3	0.675107	0.645645
	4	0.683710	0.630741
+ CutMix2D	0	0.657769	0.617192
	1	0.659733	0.635748
	2	0.648198	0.630082
	3	0.659187	0.635762
	4	0.66703	0.625469