Are whales are too big to fly?

Technical report: BioDCASE task 4 2026

Valentin Bordoux, Bram Cuyx, Clea Parcerisas, Elena Schall

June 22, 2026

1 Introduction

Passive Acoustic Monitoring (PAM) is a powerful, non-invasive tool for long-term ecosystem monitoring across both terrestrial and underwater environments. However, the scalability of PAM is currently limited by the immense effort required for manual data analysis and the scarcity of ground-truth data needed to train deep learning models for automation. Manual data annotation is exceptionally labour-intensive. To mitigate this, active learning strategies can be employed to maximize resources by selecting only the most informative samples, thereby optimizing model performance with minimal manual labelling effort. In this challenge, a sub-selection of samples from four distinct datasets (three avian and one marine mammal) is provided alongside their corresponding embeddings, which were extracted using the pre-trained Perch 2.0 model [1]. The objective is to develop an optimal sample selection strategy that maximizes downstream classification performance across these datasets under a constrained labelling budget.

2 Dataset Exploration

Before selecting specific sampling strategies, we analysed the dataset distributions. For the BioDCASE Task 4 challenge, the total labelling budget is limited to 500 samples per dataset. The datasets—HSN, POW, UUH, and ATBFL—contain 19, 45, 25, and 7 classes, respectively. Under a uniform allocation, this budget translates to a range of roughly 11 to 71 samples per class. Given this constrained budget relative to the high number of classes, we hypothesized that the success of the active learning pipeline depends mostly on capturing representative samples for all classes, rather than learning from edge cases where the model lacks confidence. Consequently, we focused our investigation on sampling strategies designed to maximize data representativeness and coverage across the embedding space, rather than usual active learning strategies based on high-uncertainty sampling methods that prioritize low-confidence boundaries.

2.1 The special case of the ATBFL dataset

The original data from the ATBFL dataset comes from the Task 2 of the bioDCASE 2025. The recordings from this task have a sampling frequency of 250 Hz. Most call categories in this dataset occur between 16 Hz and 40 Hz, but the input given to the Perch 2.0 model is a "[...] monaural audio sampled at 32 kHz and produces a log mel-spectrogram. The frontend takes in audio segments of length 5 s (160,000 samples) and uses a hop and window length of 10 and 20 ms respectively, outputting a total of 500 frames with 128 mel-scaled frequency bins ranging from 60 Hz to 16 kHz". Therefore, most of the temporal-spectral information from the whale calls in the ATBFL dataset will not be encoded in the perch embeddings. None of the call categories has significant energies at higher frequencies, except for the overtone of the 20 Hz Plus call (Bp20Plus class, at around 80 Hz) and part of the frequency range of blue whale D-Calls and fin whale 40Hz-Calls (BmD and Bp40) [2]. For this reason, we consider it unlikely that any sampling strategy based on these embeddings will significantly improve the random selection for this specific dataset.

3 Methods

We tried several sampling strategies and compared them, combined with the different warm-up strategies. Because the goal is to sample as many (equally distributed) positive samples as possible, we hypothesize that if we want to use the model's confidence as an indicator of probability of a sample belonging to a certain class, a warm-up function is necessary, as otherwise the first round of selection will be more or less random.

3.1 Warm-up Strategy

3.1.1 K-means

We used K-means clustering as a warm-up strategy to initialize the first active learning iteration. This approach ensures thorough exploration of the embedding space by selecting diverse, representative samples from each cluster. During preliminary testing, we evaluated a sample budget equal to the number of classes (aiming for one sample per class under ideal clustering conditions) and compared applying K-means directly to the high-dimensional embedding space versus a UMAP-reduced 2D space. Ultimately, for the final results, we utilized the full embedding dimensions and matched the budget used in the alternative warm-up strategies.

3.1.2 Uniform sampling after projection on the principal eigenvectors

This method aimed to sample the embedding space as uniformly as possible to best cover the input distribution of the provided dataset. This was achieved by first determining the eigenvectors and corresponding eigenvalues. The two eigenvectors with the highest eigenvalues were chosen to project the embeddings onto; as these eigenvectors explained the largest variance in the embedding space. By projecting onto them, we attained a more interpretable two-dimensional space in which a grid sampling was possible. The warm-up budget was divided between the two eigenvectors proportionally to their corresponding eigenvalues. For each of the chosen eigenvectors, a probability distribution of the projection was made. This distribution is then divided into a number of equally distributed quantiles corresponding to the assigned warm-up budget. In each of these quantiles a single sample is randomly selected, ensuring that the entire input data distribution is covered.

3.2 Sampling Strategy

3.2.1 Quantile sampling

Following a similar method to the one proposed in Navine et al. (2024) [3], we select the samples depending on the model confidence by quantiles for each class confidence value. The different quantiles are defined with these confidence limits (for all proposed strategies): [0, 0.25, 0.85, 1].

3.2.2 High-confidence single class priority

The samples chosen are the ones where the model is very confident about one class being present (higher quantile for one single class) and quite confident that no other class is present (lower quantile for all the other classes).

3.2.3 High-confidence multiple class priority

This strategy selects samples for which the model exhibits highly confident predictions across multiple classes, regardless of whether the confidence indicates class presence or absence (assuming that low confidence of presence is equivalent to high confidence of absence). For each class, prediction scores are partitioned into quantiles, and a utility score is assigned based on the number of class predictions that

fall into either the highest or lowest quantile. Samples with a larger proportion of predictions at these confidence extremes receive higher utility values.

3.2.4 At least a good sample of a class

This strategy prioritises samples for which the model predicts the presence of multiple classes with high confidence. The idea here is to increase the probability of having positive samples, regardless of which class. For each class, prediction scores are divided into quantiles, and samples receive a utility score based on the number of class predictions that fall into the highest quantile. Consequently, samples that are confidently assigned to a larger number of classes obtain higher utility values and are more likely to be selected. This approach favours instances with a higher probability for multiple positive labels.

3.2.5 Balance classes by cluster

1. Perform k-means on the normalised embedding space of all the data. This clustering is only performed at the beginning of the sampling loop
2. Each sampling iteration, perform the following (for each step, sampled samples are excluded from the sample pool for the next step):
    - 2.1 Using the historical samples, check which clusters are possibly noise clusters (no positive label). The noise clusters are the ones with more than 50% of the revealed samples classified as noise. Noise clusters are excluded from the selection in this sampling step
    - 2.2 Check from historical samples how many samples per class have been sampled so far. The median number of samples per class (above 0) is stored, and so it the maximum number of samples per class (median_n_samples and max_n_samples, respectively)
    - 2.3 Check if all the obtained k-means clusters have at least one revealed sample. If not, sample median_n_samples from each unsampled cluster
    - 2.4 Check if any class has 0 revealed samples. If that is the case, and there is still sampling budget available, sample the median_n_samples samples with a higher predicted probability of containing a positive of that class
    - 2.5 If there is still sampling budget remaining, select proportionally to the available budget the samples needed to bridge the gap per class between samples per class and the maximum number of samples per class. To this, per each class samples are randomly selected from the all the clusters which contain at least one revealed sample of that class.

3.2.6 Balance classes by confidence

The strategy aims to reduce class imbalance by allocating a larger portion of the sampling budget to classes with fewer revealed examples. For each class, unlabeled samples are ranked according to the model's predicted score, and samples where the model is the most confident of the presence of a specific class are selected in proportion to the degree of underrepresentation of that class. This preferentially acquires examples that are expected to balance class representation.

3.2.7 Sklearn CoreSet no noise

Modification of the baseline sklearn coreset function but removing as candidates all the samples from clusters where >60% of the revealed samples are noise. To determine this, first normalized embeddings are clustered using k-means algorithm, with number of clusters equal to the number of classes. At each iteration, all the clusters are evaluated, and if any cluster has more than 2 revealed samples, and >60% of these samples are noise, this cluster is excluded from the possible pool of samples for this iteration.

3.2.8 Sklearn CoreSet furthest from noise

Modification of the baseline sklearn coreset function where the sampling function greedily selects samples that are farthest from the current *noise* labelled set in embedding space, maximising coverage without considering model uncertainty. The difference with the sklearn coreset is that only noise samples are considered.

4 Results

We ran the different possible combinations of our strategies and the warm-up functions. None of the proposed sampling functions performed better than the baseline sklearn coreset, but the two proposed warmup functions improved it marginally. The proposed warmup functions did not increase performance in all strategies. Results per method are listed in Table 1.

5 Discussion

Strategies relying directly on model confidence as an estimate of class presence or absence produced weaker results than strategies relying only on the distribution of the samples in the embedding space, likely because confidence scores were not sufficient indicators during the early stages of training and therefore reinforced existing model selection biases. Given the large number of classes relative to the limited annotation budget, the primary objective becomes maximising the proportion of informative samples while minimising the selection of noise and non-target recordings. In this context, diversity-based approaches proved more effective. In particular, the scikit-learn coreset strategy produced a more balanced representation of classes, suggesting that sampling methods based on feature-space coverage are better suited to maintaining class balance under constrained labelling budgets. As expected, increasing the number of annotated samples available did not have any impact on the performance of the models on the ATBFL dataset.

6 Acknowledgements

The work of C.P was supported by the grant 1281026N from the funding agent FWO.

References

[1] Bart van Merriënboer et al. *Perch 2.0: The Bittern Lesson for Bioacoustics*. en. arXiv:2508.04665 [cs]. Aug. 2025. DOI: 10.48550/arXiv.2508.04665. URL: http://arxiv.org/abs/2508.04665 (visited on 08/20/2025).

[2] Brian S. Miller et al. "An open access dataset for developing automated detectors of Antarctic baleen whale sounds and performance evaluation of two commonly used detectors". en. In: *Scientific Reports* 11.1 (Jan. 2021). Number: 1, p. 806. ISSN: 2045-2322. DOI: 10.1038/s41598-020-78995-8. URL: https://www.nature.com/articles/s41598-020-78995-8 (visited on 07/28/2023).

[3] Amanda K. Navine et al. "All thresholds barred: direct estimation of call density in bioacoustic data". English. In: *Frontiers in Bird Science* 3 (Apr. 2024). ISSN: 2813-3870. DOI: 10.3389/fbirs.2024.1380636. URL: https://www.frontiersin.org/journals/bird-science/articles/10.3389/fbirs.2024.1380636/full (visited on 06/08/2026).

Table 1: Performance metrics by strategy and warmup method, macro-averaged by dataset. In **bold** the best performing strategies.

Strategy	**Warmup**	**mAP_mean**	**aulc_mAP_mean**
all quantiles	density	0.522262	0.416060
	eigenvalues	0.529648	0.419076
	kmeans	0.531315	0.420102
balance class by clusters	density	0.550681	0.432055
	eigenvalues	0.551395	0.434106
	kmeans	0.542808	0.430413
balance class by confidence	density	0.498690	0.392658
	eigenvalues	0.486528	0.390583
	kmeans	0.497759	0.393041
best multiclass	density	0.442192	0.367946
	eigenvalues	0.443292	0.370484
	kmeans	0.446225	0.370950
best single	density	0.440817	0.373677
	eigenvalues	0.440759	0.372846
	kmeans	0.439248	0.372504
most confident classes	density	0.440545	0.371647
	eigenvalues	0.441179	0.371653
	kmeans	0.441545	0.372219
sample from dense clusters	density	0.446346	0.330860
	eigenvalues	0.420755	0.312386
	kmeans	0.442791	0.335155
sklearn coreset	density	0.573662	0.447237
	eigenvalues	0.575550	**0.447780**
	kmeans	**0.577090**	0.447545
sklearn coreset far from noise	density	0.555433	0.439945
	eigenvalues	0.554842	0.440109
	kmeans	0.555230	0.441073
sklearn coreset no noise	density	0.551119	0.435816
	eigenvalues	0.553890	0.438899
	kmeans	0.561222	0.440940