Adaptive Foraging Learning (AFL): A Swarm-Inspired Sampling Strategy for Active Learning in Bioacoustics

BioDCASE 2026 Task 4 - Active Learning for Bioacoustics

Jaqueline Garcia Yi (Ecosonus)

Abstract

We propose Adaptive Foraging Learning (AFL), a swarm-based, bio-inspired active learning sampling strategy for BioDCASE Task 4. AFL is designed for fixed-model bioacoustic classification under a strict budget of 500 labeled samples, where the model remains unchanged and the key challenge is selecting which unlabeled audio samples to annotate. Inspired by collective foraging, AFL balances exploration and exploitation throughout the active learning loop: it uses a swarm-based warm start over the precomputed embedding space to build an initial diverse labeled set, then progressively prioritizes samples where the model is less confident while preserving acoustic diversity within each selected batch. The strategy combines embedding-space coverage, prediction uncertainty, adaptive weighting, and diversity-aware batch construction. A fine-tuned AFL configuration achieved the highest average final AULC among the evaluated strategies, including random sampling, margin-based multilabel uncertainty, CoreSet, and TypiClust, reaching 0.487118 across the evaluated datasets/subsets. Ablation experiments indicate that coverage/diversity, the swarm-based warm start, diversity-aware reranking, and adaptive weighting are the main contributors to performance, while density scoring and the front-loaded scheduler play secondary roles.

Keywords: active learning, bioacoustics, swarm intelligence, adaptive sampling, embedding coverage, uncertainty sampling, AULC, BioDCASE

1. Introduction / Overview

Large-scale bioacoustic monitoring produces more audio than can be manually annotated, making active learning essential for selecting informative samples under limited labeling budgets. BioDCASE Task 4 addresses this challenge by evaluating active learning strategies that select which unlabeled audio samples should be annotated under a fixed labeling budget. The official ranking metric is the Area Under the Learning Curve (AULC) computed from macro mean average precision (macro mAP), which rewards methods that improve model performance quickly using few labels.

In this fixed-model, budget-limited setting, the classifier architecture and training pipeline remain unchanged. The main design space is therefore the sampling strategy, meaning which unlabeled samples should be selected for annotation and in what order. Adaptive Foraging Learning (AFL) was designed for this scenario. It first explores diverse regions of the acoustic embedding space, then gradually focuses annotation effort on samples expected to provide high learning value. Model uncertainty is used as one signal, but AFL continues to preserve embedding-space diversity within each selected batch.

The method is inspired by collective foraging and swarm intelligence. In these systems, groups can search efficiently by combining exploration, information sharing, and adaptive resource allocation through simple local decision rules (Bonabeau et al., 1999; Charnov, 1976; Stephens and Krebs, 1986).

The swarm-based initial exploration phase corresponds to AFL's warm start. In biological swarms, individuals do not need a central controller to find useful regions of an environment. Instead, collective behavior emerges from multiple agents exploring different options, retaining useful information, and being influenced by successful discoveries made by others. AFL translates this principle into the warm-start phase using a discrete swarm search inspired by Particle Swarm Optimization (PSO) over candidate subsets (Kennedy and Eberhart, 1995). Unlike standard continuous PSO, where particles update numerical positions and velocities, AFL operates on discrete

sample indices and updates candidate subsets through recombination and resampling. The most diverse and representative subset found is then selected as the initial labeled set. This warm start is counted within the total annotation budget and does not use ground-truth labels or model predictions for selection.

After this initialization, AFL follows the standard active learning loop. The fixed model is trained on the labeled set, predictions are computed for the remaining unlabeled samples, and the next samples are selected using embedding-space coverage, adaptive coverage–uncertainty weighting, and diversity-aware batch construction. Early selections emphasize acoustic-space coverage, while later selections increasingly use model uncertainty to identify samples with high expected learning value.

This report describes the AFL method, its alignment with the BioDCASE Task 4 requirements, the experimental setup, baseline comparisons, ablation studies, and hyperparameter refinement.

2. Method: Adaptive Foraging Learning

Adaptive Foraging Learning (AFL) is implemented as the custom strategy in sampling.py. The method is inspired by collective foraging: it first explores the acoustic embedding space broadly, then progressively focuses annotation effort on samples expected to provide high learning value. It does not modify the classifier architecture or training pipeline, but changes how unlabeled samples are prioritized for annotation.

The complete AFL pipeline has six components:

1. **Swarm-based warm start**: selects the initial 50 samples using only the precomputed embedding space, before model predictions are available.

2. **Utility scoring**: computes three signals for each unlabeled sample after each training cycle: (a) uncertainty from model predictions, (b) coverage from distance to the current labeled set in embedding space, and (c) density from local nearest-neighbor similarity in the embedding space.

3. **Adaptive exploration-to-exploitation weighting**: starts with coverage-only selection and then switches to a hybrid coverage–uncertainty–density score.

4. **Diversity-aware batch construction**: avoids redundant batches by reranking the top utility candidates using greedy farthest-point selection.

5. **Budget-aware front-loaded scheduler**: allocates the remaining annotation budget across active learning cycles, giving more labels to earlier cycles in the submitted AFL configuration.

6. **Computational efficiency mechanisms**: use approximate nearest-neighbor search, pool subsampling, and top-candidate reranking to keep sampling cost manageable.

The main performance-driving components are the swarm-based warm start, embedding-space coverage/diversity, adaptive weighting, and diversity-aware batch construction. AFL also includes two secondary components: density scoring and a front-loaded sampling schedule. Density scoring helps reduce the selection of isolated outliers by favoring locally representative samples in the embedding space. The front-loaded scheduler allocates more labels to earlier active learning cycles, which is aligned with the AULC objective because earlier improvements contribute more to the learning curve. Ablation results in Section 5 suggest that these two components have smaller impact than coverage, warm start, adaptive weighting, and batch diversity, but they are retained in the final configuration because they are lightweight and may improve robustness.

2.1 Swarm Warm Start

Before the first active learning cycle, AFL must choose an initial set of samples without guidance from a trained model. To do this, the warm start uses only the precomputed audio embeddings. It selects samples that are spread

across the embedding space and representative of the unlabeled pool. No ground-truth labels, model predictions, or uncertainty scores are used at this stage, and the selected samples are counted within the total 500-sample annotation budget.

The warm start uses a discrete swarm-style search inspired by Particle Swarm Optimization (PSO). Multiple particles are initialized, where each particle represents a candidate subset of initial samples. Each subset is evaluated using diversity among selected samples and coverage of the candidate pool. During the internal swarm search, AFL keeps the candidate subset with the highest warm-start score found so far. At each internal iteration, each particle is rebuilt using a 40/40/20 recombination rule: 40% of the new subset is sampled without replacement from the particle's previous subset, preserving particle-level memory; 40% is sampled without replacement from the current global-best subset, incorporating swarm-level information; and the remaining 20% is newly sampled from the candidate pool using density-biased probabilities. Thus, the warm start is stochastic, but not purely random: most of each particle is guided by its previous state and by the best subset found so far, while only the refresh portion introduces new density-biased exploration.

The warm-start scoring function used to rank candidate subsets emphasizes diversity and coverage:

warm_start_score = 0.45 * diversity + 0.40 * coverage

The coefficients are used as relative weights for the two scoring terms and are not intended to form a normalized convex combination. Diversity is computed as the average pairwise squared distance among the selected samples in the embedding space. This encourages the initial labeled set to contain acoustically different examples. Coverage measures how well the selected subset spreads across the candidate pool. For each candidate-pool sample, AFL computes the distance to its nearest selected sample, then summarizes these nearest-neighbor distances using the 10th percentile. Higher coverage values indicate that the selected samples are less concentrated in one region and provide broader coverage of the embedding pool.

Density is not included as a direct term in the warm-start scoring function. Instead, it is used to bias sampling during initialization and subset refresh steps. For each candidate sample i, AFL computes the average distance to its 10 nearest neighbors, knn_dist_i, and defines density_i = exp(-knn_dist_i / mean(knn_dist)). Samples in denser regions have smaller nearest-neighbor distances and therefore higher density scores. These scores are used as sampling probabilities when initializing particles and when drawing the 20% newly sampled candidates in the 40/40/20 update rule.

AFL uses 10 particles and 10 internal iterations over a candidate pool of 800 unlabeled samples. Each particle represents a candidate subset of 50 samples, and the final warm-start output is one selected subset of 50 samples.

Final configuration:

WARM_START_SAMPLES = 50
candidate_pool_size = 800
n_particles = 10
n_iters = 10

2.2 Utility Scoring

After the warm start, AFL follows the standard active learning loop. At each cycle, the model is trained on the current labeled set and produces predictions for the remaining unlabeled samples. AFL then computes three utility signals for each unlabeled sample: uncertainty, coverage, and density.

Uncertainty is computed from the current model predictions. Because the task is multilabel, AFL estimates uncertainty from how close each predicted class probability is to the 0.5 decision boundary and from binary entropy. Predictions close to 0.5 indicate that the model is unsure whether a class is present or absent, while predictions close to 0 or 1 indicate higher confidence. These class-level uncertainty values are aggregated into a sample-level

uncertainty score, where higher values indicate that the model is less confident about the sample. Coverage is computed in the embedding space as the distance from each unlabeled sample to the current labeled set. Samples farther from already labeled anchors receive higher coverage scores, encouraging AFL to select sounds from underrepresented regions of the acoustic space.

Density is estimated from local k-nearest-neighbor similarity in the embedding space. It acts as a weak regularizer against isolated outliers, helping AFL prefer samples that are informative but still locally representative.

2.3 Adaptive Weighting and Transition Threshold

AFL uses a two-stage scoring strategy. During the early phase, when the labeled set is still small, AFL ignores model uncertainty and selects samples only by embedding-space coverage. This avoids relying too strongly on uncertainty estimates from a model trained with very few labels.

After 35% of the annotation budget has been used, AFL switches to a hybrid utility score that combines coverage, uncertainty, and density. Coverage remains the dominant signal, but uncertainty is gradually given more influence as the model becomes better trained. Density is kept as a weak regularizer to reduce the selection of isolated outliers.

The hybrid utility score is:
utility = w_unc * uncertainty + w_cov * coverage + w_den * density

Table 1. AFL Transition and Adaptive Weighting Rules

Condition	Utility used	Interpretation
Early phase: *progress* < 0.35	Coverage	Pure exploration based on embedding-space coverage.
Hybrid phase with flat uncertainty: progress >= 0.35 and uncertainty_std < 0.05	0.08 * uncertainty + 0.87 * coverage + 0.05 * density	If uncertainty scores are nearly flat, AFL treats uncertainty as unreliable and relies mostly on coverage.
Hybrid phase, normal case: progress >= 0.35 and uncertainty is informative	(0.10 + 0.20 * progress) * uncertainty + (0.75 - 0.15 * progress) * coverage + 0.03 * density	As the budget progresses, uncertainty receives more weight, while coverage remains the main signal.

Here, *progress* is the fraction of the annotation budget already used. The transition threshold *progress = 0.35* was selected through local hyperparameter tuning. This design preserves early exploration while allowing the method to incorporate model uncertainty once the labeled set is large enough to make uncertainty estimates more informative.

2.4 Diversity-Aware Batch Construction

After computing utility scores for the unlabeled samples, AFL does not directly select the top k highest-scoring samples. Direct top-k selection can produce redundant annotation batches, because many high-utility samples may be acoustically similar or located in the same region of the embedding space.

To avoid this, AFL uses a two-step batch construction procedure. First, it keeps a shortlist of high-utility candidates containing the top 5 × k samples according to the utility score. Second, AFL selects the final batch of k samples from this shortlist using greedy farthest-point selection in the embedding space. This means that each new sample added to the batch is chosen to be far from the samples already selected for that batch.

In the final configuration:
candidate_mult = 5
candidate_size = min(len(unlabeled_indices), n_samples * candidate_mult)

Thus, for a batch of k samples, AFL first considers the top 5 × k candidates and then selects a diverse subset of k samples from them. This preserves high utility while reducing redundancy within each annotation batch.

2.5 Budget-Aware Scheduler and Computational Efficiency

AFL also includes a budget-aware sampling schedule. After the 50-sample warm start, the remaining 450 labels are allocated across the 15 active learning cycles. In the submitted AFL configuration, the schedule is front-loaded: earlier cycles receive larger annotation batches, while later cycles receive smaller batches. This was intended to improve AULC, because labels acquired earlier can improve the model for more of the learning curve. However, the ablation study shows that replacing this front-loaded schedule with a flat schedule produced nearly identical average AULC, so the scheduler is treated as a secondary component (see Section 5).

To keep sampling computationally practical, AFL avoids applying expensive distance-based operations to the full unlabeled pool whenever the pool is large. Instead, coverage and density calculations are performed on an approximate sampled pool when the number of unlabeled samples exceeds the configured approximation size. Nearest-neighbor operations use approximate FAISS HNSW indices, and diversity-aware reranking is applied only to the top 5 × k utility candidates in each cycle, where k is the number of samples selected in that cycle. This keeps the sampling step focused on a small high-utility candidate set rather than repeatedly reranking the full unlabeled pool.

2.6 Task Alignment and Design Choices

AFL was designed to comply with the BioDCASE Task 4 setting. The task evaluates active learning methods that prioritize which unlabeled audio samples should be annotated using precomputed perch_v2 embeddings and model predictions. The experiments use the BioDCASE Task 4 datasets and BaseAL framework, including ATBFL and the BirdSet-derived evaluation subsets HSN, POW, and UHH (Kurinchi-Vendhan et al., 2026; Rauch et al., 2026; McEwen and Zhang, 2026). The ranking metric is the Area Under the Learning Curve (AULC) computed from macro mAP under a maximum budget of 500 labeled samples, averaged across independent runs and evaluated domains.

The implementation uses only the allowed intervention points: the warm-up sampling procedure, the batch size per cycle, and the sampling function that returns the next samples for annotation. The classifier architecture and the core active learning loop remain unchanged.

Table 2. AFL Task Alignment and Design Choices

Requirement	AFL design choice
No ground-truth labels for sampling	AFL uses only precomputed embeddings, model predictions after training, and the current labeled and unlabeled sample pools. Ground-truth labels are not used to select unlabeled samples
Fixed model and AL loop	AFL is implemented through sampling.py and notebook-level warm start / scheduling logic. model.py and active_learner.py remain unchanged.
Budget = 500 samples	The warm start selects 50 samples, counted within the total budget. The remaining 450 labels are allocated across active learning cycles.
Cross-dataset generalization	AFL relies mainly on embedding-space coverage/diversity, with uncertainty used adaptively rather than as the only signal.
Computational efficiency	AFL reduces sampling cost by using approximate nearest-neighbor search, computing coverage and density over a sampled approximation of the unlabeled pool when the pool is large, and applying diversity-aware reranking only to the top 5 × k utility candidates in each cycle, where k is the batch size.

3. Experimental Setup

The experiments were run under the BioDCASE Task 4 active learning framework using precomputed perch_v2 embeddings. The submitted method name remains custom to match the BaseAL framework. AFL was compared

against four baseline strategies: random, margin_multilabel, sklearn_coreset, and sklearn_typiclust, using the same fixed model and active learning setup.

Table 3. Experimental Configuration

Parameter	Value
Datasets	ATBFL, HSN, POW, UHH
Embedding model	perch_v2
Compared strategies	random, margin_multilabel, sklearn_coreset, sklearn_typiclust, custom / AFL
Independent repeats	5 full-run repeats
Active learning cycles	15
Epochs per cycle	10
Training batch size	32
Learning rate	0.001
Maximum annotation budget	500 labeled samples
Warm start	50 samples, applied only to AFL / custom
Sampling schedule	Front-loaded scheduler for AFL / custom; flat budget allocation for baselines
Main metric	AULC based on macro mAP
Development device	CPU

Each repeat used a fresh ActiveLearner instance with a new random model initialization. Results were aggregated using the average final AULC across datasets and repeats.

4. Results

AFL was evaluated against the baseline sampling strategies included in the framework: random, margin_multilabel, sklearn_coreset, and sklearn_typiclust. The main metric is the average final AULC based on macro mAP under the 500-sample annotation budget.

4.1 Global Ranking by Average Final AULC

The final AFL configuration achieved the highest average final AULC among the evaluated strategies. Compared with the strongest baseline, sklearn_coreset, AFL improved average final AULC by 0.015948 absolute points, corresponding to a 3.39% relative improvement. Compared with random sampling, AFL improved average final AULC by 0.083791 absolute points, corresponding to a 20.78% relative improvement.

Table 4. Final AULC Comparison Across Strategies

Strategy	Average final AULC
custom / AFL	0.487118
sklearn_coreset	0.471170
sklearn_typiclust	0.420620
margin_multilabel	0.415251
random	0.403327

4.2 Detailed Results by Dataset

AFL achieved the highest final AULC on all evaluated datasets/subsets. On UHH, CoreSet achieved slightly higher final mAP, while AFL achieved higher AULC, indicating faster learning across the budget.

Table 5. Detailed Results by Dataset/Subset

Dataset	Strategy	Final mAP	Final AULC
ATBFL	custom / AFL	0.509976	0.472371
ATBFL	random	0.503977	0.462933
ATBFL	sklearn_coreset	0.492708	0.456792

ATBFL	sklearn_typiclust	0.493738	0.455608
ATBFL	margin_multilabel	0.476338	0.435440
HSN	custom / AFL	0.697232	0.571839
HSN	sklearn_coreset	0.653606	0.542374
HSN	margin_multilabel	0.610731	0.453689
HSN	sklearn_typiclust	0.475529	0.408073
HSN	random	0.471624	0.368981
POW	custom / AFL	0.630209	0.479544
POW	sklearn_coreset	0.591295	0.467950
POW	sklearn_typiclust	0.580720	0.459788
POW	random	0.569861	0.444596
POW	margin_multilabel	0.607128	0.441885
UHH	custom / AFL	0.533903	0.424719
UHH	sklearn_coreset	0.537565	0.417565
UHH	sklearn_typiclust	0.425469	0.359011
UHH	random	0.406874	0.336800
UHH	margin_multilabel	0.440232	0.329990

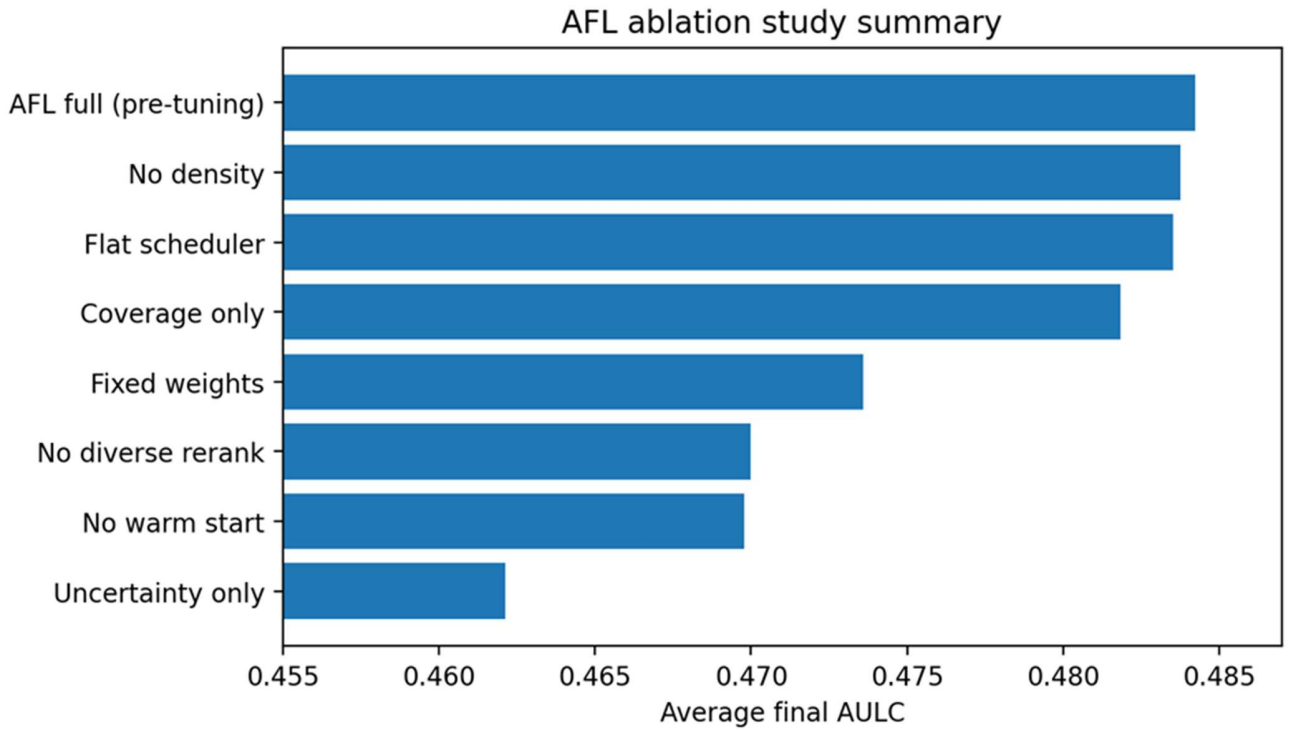
5. Ablation Study

The ablation study tested whether AFL performance is driven by specific components or by the full combination. Each ablation removed or isolated one design element and compared its average final AULC against the full AFL reference configuration before hyperparameter fine-tuning.

Table 6. Ablation Study Results Relative to Pre-Tuning AFL Configuration

Experiment	Avg final AULC	Delta vs AFL full
AFL full (pre-tuning)	0.484213	0.000000
AFL no density	0.483758	-0.000455
AFL flat scheduler	0.483519	-0.000694
AFL coverage only	0.481841	-0.002372
AFL fixed weights	0.473585	-0.010628
AFL no diverse rerank	0.469976	-0.014237
AFL no warm start	0.469774	-0.014439
AFL uncertainty only	0.462131	-0.022082

Figure 1. AFL Ablation Study Summary.



The strongest conclusion is that AFL is primarily driven by embedding-space coverage and diversity. Coverage-only retains most of the full method performance, while uncertainty-only drops substantially. However, the full method still performs best, indicating that model uncertainty and adaptive weighting provide useful refinements once an initial diverse labeled set has been established. Removing density and replacing the front-loaded scheduler with a flat scheduler produced nearly identical average AULC in these runs, suggesting that both components are secondary rather than essential to AFL performance.

Table 7. Interpretation of Ablation Results

Component	Evidence	Importance
Embedding coverage/diversity	Coverage-only remains close to the full method.	Very high
Model uncertainty alone	Uncertainty-only has the lowest AULC among AFL variants.	Insufficient alone
Diversity-aware reranking	Removing reranking causes a large AULC drop.	High
Swarm warm start	Removing warm start causes a large AULC drop.	High
Adaptive weighting	Fixed weights reduce performance.	Medium-high
Density scoring	Removing density produced nearly identical average AULC in these runs.	Low
Front-loaded scheduler	Using a flat scheduler produced nearly identical average AULC in these runs.	Low

The ablations also show that batch construction matters. Removing diversity-aware reranking causes a clear performance drop, suggesting that selecting high-utility samples directly can create redundant annotation batches. Similarly, removing the swarm warm start reduces AULC, supporting the value of an initial embedding-based exploration phase.

6. Hyperparameter Fine-Tuning

Hyperparameter refinement was performed locally around the AFL configuration rather than as an exhaustive search. The tuning focused on parameters most directly related to the ablation findings: warm-start size, diversity reranking candidate multiplier, coverage-to-hybrid transition timing, and uncertainty/coverage weighting.

Table 8 reports development runs used to choose the final AFL configuration. The final comparison with all strategies is reported separately in Section 4.

Warm-start size and candidate-multiplier variants were screened with one repeat to eliminate clearly underperforming settings. The most promising transition-threshold variants were then evaluated with five repeats.

Table 8. Hyperparameter Fine-Tuning Results Used for Final AFL Configuration Selection

Changed parameter	Avg final AULC	Interpretation
Reference AFL configuration: progress_switch = 0.45	0.484213	Baseline AFL configuration before tuning
Warm start = 25	0.480480	Did not improve; insufficient initial exploration. One-repeat screening result.
Warm start = 75	0.459138	Did not improve; consumes too much budget before model-guided sampling. One-repeat screening result.
Candidate multiplier = 3	0.475000	Did not improve; candidate pool too small for diverse batches. One-repeat screening result.
Candidate multiplier = 8	0.479600	Did not improve; excessive diversity may sacrifice utility. One-repeat screening result.
Progress switch = 0.55	0.485299	Improved over reference across five repeats, but below switch35.
Progress switch = 0.35	0.485649	Best confirmed tuning result across five repeats.
Switch35 + more coverage	0.484188	No clear improvement over switch=0.35 with original weights.
Switch35 + more uncertainty	0.485256	Slightly below switch=0.35 with original weights.

Note: Warm-start size and candidate-multiplier variants were screened with one repeat, while transition-threshold variants were confirmed with five repeats.

The only hyperparameter that consistently improved performance was the transition threshold. Moving the switch from progress_switch = 0.45 to progress_switch = 0.35 increased average AULC from 0.484213 to 0.485649 across five repeats. This suggests that AFL benefits from an initial coverage-driven phase, but model uncertainty should be incorporated relatively early once the warm start has created a basic diverse labeled set.

The final method keeps WARM_START_SAMPLES = 50 and candidate_mult = 5, retains the original adaptive uncertainty/coverage weights, and updates the transition threshold to progress_switch = 0.35.

7. Final AFL Configuration

Final AFL combines the strongest-performing components identified by ablation and tuning: swarm-based warm start, embedding-space coverage, adaptive weighting, and diversity-aware batch construction. Not all parameters in the final configuration were tuned. The tuning stage selected the warm-start size, diversity candidate multiplier, and coverage-to-hybrid transition threshold. Other implementation parameters, including the warm-start candidate pool size, number of particles, and number of internal iterations, were fixed to keep the search computationally efficient.

Table 9. Final AFL Configuration

Parameter	Final value/ decision
Warm start samples	50
Warm-start candidate pool	800 unlabeled samples

Warm-start particles	10
Warm-start internal iterations	10
Diversity candidate multiplier	5
Coverage-to-hybrid switch	progress_switch = 0.35
Adaptive weights	Original adaptive uncertainty/coverage/density weights retained
Density term	Retained in full AFL, although ablations show it is secondary
Batch construction	Diversity-aware reranking retained
Scheduler	Front-loaded scheduler retained in the full configuration, although ablations show it is secondary

After selecting this configuration through local tuning, we re-ran the final comparison against the baseline strategies. In that final evaluation, AFL achieved an average final AULC of 0.487118. This is the final reported AFL result; the tuning results in Section 6 were used only to choose the final configuration.

8. Discussion

The main empirical finding is that AFL is primarily a coverage-driven active learning method, not a pure uncertainty sampler. This is consistent with active learning literature showing that sample selection should balance informativeness and representativeness, especially under limited labeling budgets (Settles, 2009). This is well aligned with bioacoustic monitoring, where labeled data are scarce, soundscapes are heterogeneous, and acoustic domains can vary strongly across sites. In these conditions, uncertainty estimates from a lightly trained model can be unstable, and pure uncertainty sampling may over-focus on ambiguous or redundant regions. AFL mitigates this by first selecting diverse and representative samples in the embedding space, then using model uncertainty as a secondary signal once useful labeled anchors exist.

The ablation results support this interpretation. The coverage-only variant retained most of the full method's performance, while the uncertainty-only variant dropped substantially. However, the full AFL method still performed best, indicating that uncertainty and adaptive weighting provide useful refinements when combined with strong embedding-space coverage.

AFL also demonstrates that batch construction matters. Selecting the highest-scoring samples directly can produce redundant batches, especially when many high-utility samples come from the same acoustic region. Diversity-aware reranking reduces this redundancy by preserving utility while improving coverage within each selected batch. This is especially important under a fixed annotation budget, where redundant annotations directly reduce label efficiency as measured by AULC.

The method is also computationally practical. AFL avoids exhaustive distance operations over the full unlabeled pool by using approximate nearest-neighbor search, limiting expensive coverage and density calculations to a sampled unlabeled pool, and applying diversity-aware reranking only to the top 5 × k utility candidates in each cycle, where k is the batch size.

9. Limitations and Future Work

- **Limited hyperparameter search.** Hyperparameter tuning tested a selected set of variants around the AFL reference configuration rather than all possible parameter combinations. Therefore, the final configuration should be interpreted as a strong locally tuned configuration, not necessarily as the globally optimal AFL design. Future work could explore a broader hyperparameter search, including warm-start search budget, adaptive weights, candidate-pool size, and scheduler variants.

- **One-repeat screening for some tuning candidates.** Some warm-start size and candidate-multiplier variants were screened with one repeat as a computationally efficient way to discard clearly underperforming configurations. This makes those screening results less robust than the five-repeat transition-threshold experiments. Future work could confirm the most relevant discarded variants with additional repeats.
- **Limited estimate of variability in final results.** The final comparison used five independent repeats, which provides only a limited estimate of run-to-run variability. Therefore, results should be interpreted as empirical performance trends rather than formal statistical significance. Future work should report standard deviations or confidence intervals over more repeats.
- **No independent held-out test evaluation.** The final configuration was selected using the available development datasets and was not evaluated on a fully independent held-out test dataset. Therefore, its generalization to unseen bioacoustic datasets remains to be confirmed. Future work should validate AFL on additional independent datasets and soundscape conditions.
- **Fixed warm-start search budget.** The internal warm-start search budget, including candidate pool size, number of particles, and number of internal iterations, was fixed rather than tuned. This was a deliberate choice to limit computational cost, but different swarm-search budgets may affect the balance between sampling quality and runtime. Future work could evaluate whether larger or adaptive swarm budgets improve performance enough to justify the additional cost.
- **Secondary role of density and scheduling.** The density term and front-loaded scheduler were retained in the final AFL configuration, but ablation results suggest that they are secondary components rather than primary performance drivers. Future work could test a simplified AFL variant without density scoring or with a flat scheduler to reduce implementation complexity.
- **Limited subdomain analysis.** The current evaluation reports aggregate performance across the available Task 4 datasets/subsets. Future work could analyze per-subdomain behavior more deeply, especially for location-specific BirdSet and ATBFL subsets, to understand when AFL benefits most from coverage, uncertainty, warm start, or batch diversity.

10. Conclusion

Adaptive Foraging Learning is designed to improve label efficiency by combining a swarm-based warm start, embedding-space coverage, adaptive weighting, and diversity-aware batch construction. Ablation studies show that coverage/diversity is the dominant signal, while the warm start and batch-level diversity are key contributors. Hyperparameter tuning further suggests that the timing of the transition from coverage-only exploration to hybrid sampling is the most sensitive design choice.

In the evaluated runs, the final AFL configuration achieved the highest average final AULC among the compared strategies, reaching 0.487118 across the evaluated datasets/subsets. This corresponds to a 3.39% relative improvement over the strongest baseline and a 20.78% relative improvement over random sampling. AFL also achieved the highest final AULC on each individual dataset/subset: ATBFL, HSN, POW, and UHH. These results support AFL as a promising and computationally practical active learning sampling strategy for BioDCASE Task 4.

References

Bonabeau, E., Dorigo M. & Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems.* New York, NY: Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity.

Charnov, E. (1976) *Optimal Foraging, The Marginal Value Theorem*. Theoretical Population Biology Vol 9, Issue 2, Pages 129-136.

Kennedy, J. and Eberhart, R. (1995). *Particle Swarm Optimization.* Proceedings of ICNN'95 - International Conference on Neural Networks.

Kurinchi-Vendhan, R., Zhang, S., & McEwen, B. (2026). *BioDCASE 2026 Task 4: ATBFL Dataset*. Zenodo. https://doi.org/10.5281/zenodo.19133111

McEwen, B., & Zhang, S. (2026). *BaseAL: Active Learning Baseline (v1.1.1).* Zenodo. https://doi.org/10.5281/zenodo.18467564

Rauch, L., Herde, M., & McEwen, B. (2026). *BioDCASE 2026 Task 4: BirdSet Dataset.* Zenodo. https://doi.org/10.5281/zenodo.19191602

Settles, B. (2009). *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin-Madison.

Stephens, D. & Krebs, J. (1986) *Foraging Theory.* Monographs in Behavior and Ecology.