

Safe Rarity-Aware k-Center Active Learning for Bioacoustic Classification

Xiangwen (Wayne) Yang — Monash University BioDCASE 2026,

Task 4 — Active Learning for Bioacoustics. Method tag: **safe_rarity**.

Abstract

We present an active-learning acquisition function for bioacoustic classification on top of frozen Perch v2 embeddings. The sampler combines three rank-normalised signals — top-k binary entropy, k-center diversity to the labeled set, and predicted-prevalence rarity — and weights the rarity term through an adaptive gate **gamma** driven by budget phase, prediction separation, and effective number of classes. Selection runs greedy score-plus-diversity over a small union pool of top-utility and top-diversity candidates. On the four BioDCASE tracks the method reaches a mean final AULC-mAP of **0.491**, a +3.2 absolute point improvement over the strongest baseline (CoreSet, 0.458) at **~6x lower sampling time per cycle**, with no change to the training loop.

1. Method

Let $\mathbf{x}_u$ be the L2-normalised Perch v2 embedding of an unlabeled candidate and $\mathbf{p}_u \in [0, 1]^C$ the current classifier's per-class probability. The sampler builds three rank-normalised signals: an **uncertainty** score $U(u)$ equal to the mean of the top-5 per-class binary entropies of $\mathbf{p}_u$, multiplied by a positive-presence gate $0.35 + 0.65 \cdot \text{pos}(u)$ when the prediction distribution has enough head/tail separation ($(q_{90}-q_{50})/q_{90} \geq 0.15$); a **diversity** score $D(u)$ equal to the rank-normalised cosine distance from $\mathbf{x}_u$ to its nearest labeled neighbour; and a **rarity** score $R(u) = \sum_c \mathbf{p}_u(c) \cdot \mathbf{w}_c$ with $\mathbf{w}_c = \text{clip}((1/\sqrt{\text{prevalence}_c})/\text{median}, 1, 2.5)$, re-ranked by $R(u) \cdot \sqrt{\text{pos}(u)} \cdot (0.5 + 0.5 \cdot U(u))$ so only confidently rare predictions contribute.

1.1 Adaptive gating

With **phase** measuring the fraction of budget past warm-up, **sep** a prediction-separation factor, **eff** a function of the effective number of classes, and **safe** the maximum of pool, concentration, and early-cycle pressure, the weights and pre-score are

$$\begin{aligned} \text{gamma} &= (1 - \text{safe}) \cdot \text{phase} \cdot \text{sep} \cdot \text{eff}, & \mathbf{w}_R &= 0.05 \cdot \text{gamma}, \\ \mathbf{w}_D &= 0.32 - \mathbf{w}_R / 3, & \mathbf{w}_U &= 1 - \mathbf{w}_D - \mathbf{w}_R, \\ \text{pre}(u) &= \mathbf{w}_U \cdot U_{\text{gated}}(u) + \mathbf{w}_D \cdot D(u) + \mathbf{w}_R \cdot R(u). \end{aligned}$$

Whenever predictions are degenerate or the budget is small, **gamma** $\rightarrow 0$ and the sampler reduces to entropy-filtered k-center — the "safe" in the method name, empirically the single biggest contributor to

AULC stability across datasets. On cycle 0 with no predictions, the sampler bypasses the pipeline and runs greedy farthest-first over a random unlabeled subset.

1.2 Candidate pool and batch selection

We assemble a candidate pool as the union of the top pre-score indices, the top diversity indices, and (when $\gamma > 0.25$) the top indices with both positive-gate and uncertainty above their 60th/40th percentiles, capped at 6000 entries. Inside the pool a greedy selector picks $\text{argmax}((1-w) \cdot \text{pre} + w \cdot \text{min_cos_dist_to_picked})$ with w ramping from $0.50 + 0.10 \cdot \text{safe}$ to 0.85 along the batch, encouraging later picks to spread.

2. Experimental Setup

All experiments use the BaseAL pipeline unmodified. Classifier: a linear head on frozen Perch v2 embeddings (~2.4M params), trained with Adam ($\text{lr} = 1e-3$) for 10 epochs per cycle, batch 32. Budget: 10 cycles of 50 samples each (500 labels total), 5 outer repeats per dataset. We report mean and SD across the four BioDCASE tracks (HSN, POW, ATBFL, UHH).

3. Results

3.1 Final AULC-mAP (mean over 5 repeats)

Method	HSN	POW	ATBFL	UHH	Mean
Random	0.482	0.426	0.430	0.232	0.392
Margin (multilabel)	0.488	0.439	0.430	0.260	0.404
TypiClust	0.494	0.461	0.444	0.265	0.416
CoreSet (k-center)	0.508	0.453	0.483	0.390	0.458
safe_rarity (ours)	0.587	0.477	0.490	0.409	0.491

Our sampler wins on every dataset; per-dataset gains over CoreSet are HSN +0.079, POW +0.024, UHH +0.019, ATBFL +0.007 (mean +0.032).

3.2 Computational cost

Per-cycle sampling wall-clock time (mean over 5 repeats):

Dataset	safe_rarity (s)	CoreSet baseline (s)	Speedup
HSN	0.068	0.326	4.8×
POW	0.046	0.166	3.6×
ATBFL	0.072	0.507	7.1×
UHH	0.108	0.764	7.1×
Mean	0.074	0.441	6.0×

`relative_cost = 1.0` since the training schedule is identical to the baselines; sampler cost is dominated by one nearest-cosine pass and one greedy diversity loop over the 6k-row pool.

3.3 Ablations

Three single-factor ablations under the same 500-budget / 5-repeat protocol:

Configuration	HSN	POW	ATBFL	UHH	Mean
safe_rarity (full)	0.587	0.477	0.490	0.409	0.491
margin only (no diversity, no rarity)	0.549	0.466	0.492	0.358	0.466
entropy only (no diversity, no rarity)	0.545	0.464	0.484	0.348	0.460
entropy + rarity, no k-center diversity	0.530	0.442	0.482	0.338	0.448

The k-center diversity component is the dominant contributor: removing it while keeping entropy and rarity costs -0.043 mean AULC, with the largest drops on HSN (-0.058) and UHH (-0.071) — the tracks with the most skewed label distributions.

4. Discussion

Two design choices drive the gain over CoreSet. All scores are rank-normalised before mixing, so dataset-specific magnitude differences do not distort the weighting; and the rarity term is gated rather than always-on, which avoids damage on cold-start cycles and on tracks with flat class distributions. The adaptive `gamma` lets the same configuration generalise across all four tracks without per-dataset tuning. A natural extension is to combine the safety gates with a modest dimensionality-reduction step over the labeled set to better resolve classes that share a single Perch axis.

Reproducibility. The submitted YAMLs were produced from `BaseAL/core/utils/sampling.py` at commit `8fc508d`; the submitted `sampling.py` is a self-contained subset of that file with `safe_rarity` as the default `BASEAL_CUSTOM_VARIANT`.