# WHICH SAMPLES TO LABEL FIRST? TESTING FAILED PICKS AS DISTANCE ANCHORS FOR RARE-CLASS DISCOVERY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Annotators searching image archives for rare species, from drone surveys of forest canopy to camera traps, often open frames that cannot be given any class label, such as empty ones, which costs budget; we call this a failed pick. A failed pick gives no class label, but remembering where it lies can still help find rare classes. Anchored $k$-center CoreSet keeps each failed pick as a distance anchor, lowering nearby images' priority without removing them. In preregistered tests on iWildCam and Snapshot Serengeti, anchoring raises discovery AUC, distinct rare classes found summed over ten rounds, by +9.5% and +32.3% over skipping alone. CoreSet that skips failed picks also beats random selection that skips them; all 16 seeds favor each of the four comparisons (exact sign-flip $p = 2^{-16}$ on each). This confirms one of three preregistered hypotheses. The two broader ones are not confirmed: the pooled distance-versus-density contrast misses significance and reverses on one of four datasets, and under re-proposal CoreSet stays ahead of random on both camera-trap datasets, opposite to the prediction. In exploratory drone-pool experiments with simulated failures, the pool convention, how a tool treats unlabelable items, moves CoreSet from 27% below random to 46% above, and an audit of 19 tools finds no common convention. On that pool, learned labelability filtering avoids more failed picks yet has lower discovery AUC than anchoring on two encoders; the third shows no significant difference in this endpoint. In an exploratory MegaDetector comparison, detector filtering raises mean discovery AUC for all three selectors but makes some rare classes unreachable; it reverses the anchoring advantage on iWildCam but not on Serengeti. We contribute evidence, not a new CoreSet algorithm, and release `labelfirst`, a library that audits rare-class discovery under each pool convention.

## 1 INTRODUCTION

An annotation team searching an image archive for rare species pays for every frame it inspects. Many frames, such as empty ones, return no class label. We call such a frame an *unlabelable item* and an inspection of one a *failed pick*. The goal is *rare-class discovery*: to see as many distinct rare classes as early as possible, not to maximize classifier accuracy. It matters in long-tailed ecological collections, where a few common classes dominate the images (ter Steege et al., 2013; Van Horn & Perona, 2017; Van Horn et al., 2018; Joly et al., 2020; Tuia et al., 2022; Pollock et al., 2025).

After a failed pick, a selector can propose the item again, skip it, or skip it and remember where it lies. A failed pick gives no class label, but remembering where it lies can still help find rare classes. We study this with greedy $k$-center CoreSet (Sener & Savarese, 2018), which picks the image farthest from its *distance anchors*, the items it treats as already covered. $\text{CoreSet}_{\text{xf}}$ skips failed picks and anchors only on labeled items. Anchored CoreSet, $\text{CoreSet}_{\text{af}}$ (arm L in our protocol), also keeps each failed pick as an anchor. Images near a failed pick then look already covered, so they lose priority but remain eligible. $\text{CoreSet}_{\text{af}}$ is not a new selection algorithm: the CoreSet in our selector registry makes identical selections when it skips inspected items.

Exploratory experiments on drone imagery of the Barro Colorado Island canopy suggested three hypotheses, which we preregistered and tested on held-out datasets. We preregistered three hypothe-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

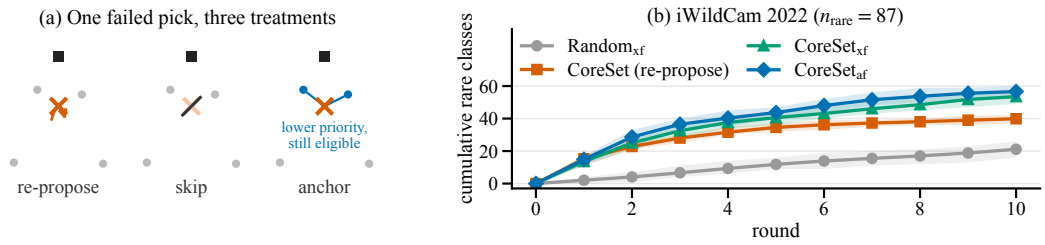



Figure 1: **Skip failed picks, anchor on them, and seek distance.** (a) One failed pick under three treatments. A loop that re-proposes it can pick it again (NAIVE). A loop that skips it never proposes it again (SELECTABLE). CoreSet$_{af}$ also anchors on it, lowering nearby candidates' priority while keeping them eligible. (b) Distinct rare classes found per round on iWildCam 2022 (87 rare classes), mean over 16 seeds with a min-max band. By round 10, Random$_{xf}$ finds 21.1, CoreSet that re-proposes finds 39.9, CoreSet$_{xf}$ finds 53.5 and CoreSet$_{af}$ finds 56.6.

ses. H2, the narrowest, says that CoreSet$_{af}$ beats CoreSet$_{xf}$ and that CoreSet$_{xf}$ beats Random$_{xf}$, random selection that also skips failed picks. Both parts hold on both camera-trap datasets (Figure 2). Anchoring raises discovery AUC, the distinct rare classes found summed over rounds, by +9.5% on iWildCam 2022 and +32.3% on Snapshot Serengeti. In secondary comparisons, CoreSet$_{af}$ leads most other skipping selectors, but LOF leads on Serengeti in point estimate. Two broader hypotheses are not confirmed. H1 predicted that distance-seeking selectors beat density-seeking ones when every item is labelable; the pooled contrast misses $\alpha$ ($p = 0.045$) and reverses on one of four datasets. H3 predicted that re-proposing failed picks would put CoreSet below random; within ten camera-trap rounds it does not.

The paper makes four contributions. **(1)** A preregistered test confirms H2 on two camera-trap datasets (Section 3). **(2)** The *pool convention*, how a tool treats unlabelable items, is a hidden variable. On an exploratory drone pool with simulated failures, it moves CoreSet from $-27.23\%$ to $+46.26\%$ against random, and an audit of 19 tools finds no common convention (Section 4). **(3)** Avoiding failed picks is not the same as finding rare classes. On the drone pool, a learned labelability filter avoids more failed picks yet discovers fewer rare classes than anchoring on two of three encoders. An exploratory MegaDetector filter raises discovery for all three tested selectors but makes some rare classes unreachable; it reverses the anchoring advantage on iWildCam but not on Serengeti (Section 4.3). **(4)** The open-source `labelfirst` library runs selectors under each pool convention and audits rare-class discovery with fixed statistical tests and per-seed evidence. It ships the simulator, selectors, audit and a provenance envelope; any selector can be wrapped in its `Strategy` protocol. Our scientific contribution is evidence, not a new CoreSet algorithm, and it concerns discovery, not classifier accuracy.

## 2 SELECTION PROCEDURE AND DISCOVERY ENDPOINT

Let $X \in \mathbb{R}^{n \times d}$ be $L^2$-normalized embeddings of $n$ images. Some images carry a class $y_i \in \mathcal{C}$. The *rare class set* $\mathcal{R} \subset \mathcal{C}$ holds the classes with at most 5 labelable examples in the full pool. A selector $\pi$ takes $X$ and the labeled indices and returns $k$ unlabeled indices. A *round* is one call and its labels. Each run starts from a randomly inspected *initial pool* and runs $K$ rounds of $k$ picks. Let $u_r$ be the number of distinct rare classes among images acquired in rounds 1 to $r$; the initial pool guides selection but is not counted. The *discovery AUC* of a (selector, seed) run is $\sum_{r=1}^{K} u_r$. It rewards early discovery, which the final count $u_K$ ignores; we report $u_K$ and the early AUC $u_1 + u_2 + u_3$ as secondary endpoints. The early AUC is the cold-start endpoint. Rarity is pool-relative. This registered endpoint credits reacquisition of an initially observed class. An exploratory sensitivity excludes such classes (Appendix A).

**Pool conventions.** A pool can hold *unlabelable items*, images that return no class label when picked, such as empty frames or images never sent for labeling. A pick of one is a *failed pick*. A simulator can treat them in four ways (Table U.1). NAIVE: the items stay in the pool, and a failed pick spends budget, returns no label and can be re-proposed, as with scikit-activeml's default

candidates (Herde et al., 2025). SELECTABLE: as NAIVE, but a failed pick is never a candidate again. COVERED: the items are passed to the selector as if labeled, so they are never picked but still shape its geometry, for example as CoreSet's distance anchors. REMOVED: the items are dropped, and every selector works on the labelable images alone. Only NAIVE and SELECTABLE are open to a real loop, which cannot know in advance which items will fail.

**Selection and budget.** CoreSet greedily picks the candidate with greatest minimum cosine distance to its anchors (Sener & Savarese, 2018; Gonzalez, 1985). Each pick becomes a temporary anchor for the rest of that batch. $\text{CoreSet}_{\text{xf}}$ retains only successfully labeled items as anchors between rounds; $\text{CoreSet}_{\text{af}}$ also retains failed picks. Both skip all inspected items, charge every pick and continue after a wholly failed round. Paired H2 arms use the same initial pool, drawn from all images; its inspection cost is part of the budget. Failures in that initial pool also anchor $\text{CoreSet}_{\text{af}}$. $\text{Random}_{\text{xf}}$ samples uninspected items uniformly. Anchoring changes distances, not eligibility: nearby candidates can still be picked. Once a successful anchor exists, $\text{CoreSet}_{\text{af}}$ treats successful and failed inspections identically: with fixed embeddings and initialization its farthest-first order does not depend on subsequent outcomes. The implementation takes an exploratory random pick when no successful anchor exists. H2a tests retention of inspected locations, not learning a failure model. The full selector catalogue is in Appendix B.

## 3 A PREREGISTERED TEST ON CAMERA TRAPS

### 3.1 DESIGN

The protocol (supplement) fixes the three hypotheses, the datasets, selectors, endpoint and tests before any confirmatory run. H2 and H3 use camera-trap frames from iWildCam 2022 (Beery et al., 2021) and Snapshot Serengeti (Swanson et al., 2015), 15,000 frames each, 46% and 76% of them unlabelable, so picking one returns no class label. The protocol counts as unlabelable empty frames, unknown birds or bats on iWildCam, and empty frames or fire on Serengeti. H1 adds PlantNet-300K (Garcin et al., 2021) and an iNaturalist Nepenthaceae slice; the tables call the four datasets D1 to D4, in this order. All four use DINOv3 ViT-L/16 embeddings, 16 seeds and ten rounds of $k = \max(10, \text{round}(N_{\text{lab}}/40))$ picks, with $N_{\text{lab}}$ labelable frames, after an initial pool of $k$ (201 on iWildCam, 88 on Serengeti). The pool conventions are those of Section 2. Each test is an exact sign-flip test on the per-seed log ratio $d$ of discovery AUC plus one, one-sided at $\alpha = 0.025$, with effects reported as $\exp(\bar{d}) - 1$.

H2 runs under SELECTABLE on the two datasets with at least 10% unlabelable items. It says that (a) $\text{CoreSet}_{\text{af}}$ beats $\text{CoreSet}_{\text{xf}}$, and (b) $\text{CoreSet}_{\text{xf}}$ beats $\text{Random}_{\text{xf}}$. $\text{CoreSet}_{\text{xf}}$ is cosine farthest-first over the uninspected pool, with the labeled items as anchors. $\text{Random}_{\text{xf}}$ never draws an inspected item again. $\text{CoreSet}_{\text{af}}$ also treats each failed pick as a distance anchor. H2a thus tests anchoring against skipping alone, as in Proposition 3(b). H1 says that under REMOVED the distance-seeking selectors

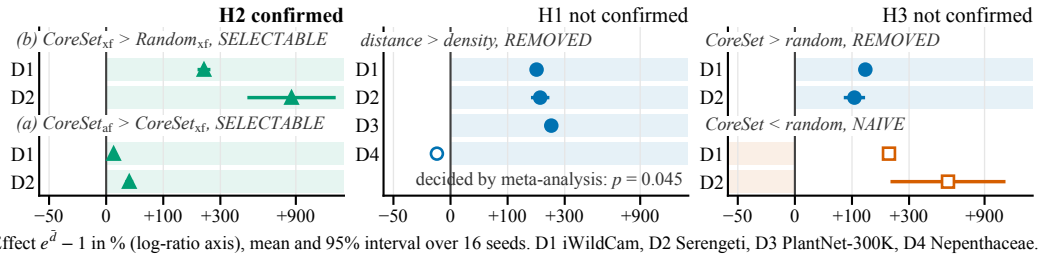



Figure 2: **H2 is confirmed on both camera-trap datasets; H1 and H3 are not.** One row per preregistered test component: mean per-seed log ratio $d$ of discovery AUC plus one over 16 seeds with a Student-$t$ 95% interval; ticks show the effect $e^{\bar{d}} - 1$. Filled markers reject in the preregistered direction at $\alpha = 0.025$ (Holm for H2), hollow ones do not; shading marks the predicted side. H1 is decided by a random-effects meta-analysis ($p = 0.045$), so three rejecting datasets do not confirm it. H3 predicted CoreSet below random under NAIVE, but it finds more rare classes there.

Table 1: **Preregistered tests: H2 confirmed, H1 and H3 not.** Discovery AUC, DINOv3, $n_s = 16$. Effect is $\exp(\bar{d}) - 1$ for the mean per-seed log ratio $d$ of discovery AUC plus one, so it differs from the ratio of mean AUCs. $p$ is the one-sided exact sign-flip $p$ in the hypothesis's direction (for H3 under NAIVE, CoreSet below random), or the random-effects $p$ for the meta row (REML: restricted maximum likelihood (Viechtbauer, 2005); HKSJ: Hartung-Knapp-Sidik-Jonkman (Hartung & Knapp, 2001; Sidik & Jonkman, 2002; Röver et al., 2015)). H1 contrasts the distance-seeking and density-seeking families under REMOVED; H2 runs under SELECTABLE. Mean AUC: each side's mean discovery AUC over seeds, for H1 averaged over the family's selectors. "reject"/"retain": each component's decision at 0.025, after Holm for H2; H1 is decided by the meta row. IUT: intersection-union verdict.

| Hyp. | Dataset | Contrast | Mean AUC | Effect | $p$ | Result |
|---|---|---|---|---|---|---|
| H1 | D1 iWildCam | $F_{\text{dist}}$ vs $F_{\text{dens}}$ | 399 / 139 | +184.2% | $2^{-16}$ | — |
| | D2 Serengeti | | 82 / 29 | +196.9% | $2^{-16}$ | — |
| | D3 PlantNet | | 1063 / 310 | +239.8% | $2^{-16}$ | — |
| | D4 Nepenthaceae | | 140 / 165 | −15.2% | 1.000 | — |
| | meta | REML, HKSJ | | +122.0% | 0.045 | not confirmed |
| H2a | D1 iWildCam | $\text{CoreSet}_{\text{af}}$ vs $\text{CoreSet}_{\text{xf}}$ | 429 / 392 | +9.5% | $2^{-16}$ | reject |
| | D2 Serengeti | | 71 / 54 | +32.3% | $2^{-16}$ | reject |
| H2b | D1 iWildCam | $\text{CoreSet}_{\text{xf}}$ vs $\text{Random}_{\text{xf}}$ | 392 / 120 | +227.9% | $2^{-16}$ | reject |
| | D2 Serengeti | | 54 / 7 | +849.1% | $2^{-16}$ | reject |
| | all | IUT | | | | confirmed |
| H3 | D1 iWildCam | CoreSet vs Random, REMOVED | 470 / 200 | +135.2% | $2^{-16}$ | reject |
| | D2 Serengeti | | 85 / 42 | +106.0% | $2^{-16}$ | reject |
| | D1 iWildCam | CoreSet vs Random, NAIVE | 323 / 103 | +213.4% | 1.000 | retain |
| | D2 Serengeti | | 42 / 10 | +541.5% | 1.000 | retain |
| | all | IUT | | | | not confirmed |

Table 2: **Exploratory: all selectors under NAIVE and SELECTABLE.** Mean discovery AUC over the preregistered seeds on the camera-trap pools. Bold: SELECTABLE beats NAIVE (exact sign-flip test, Holm across the non-random selectors). Initialization and stopping differ between conventions (Appendix T). $\text{CoreSet}_{\text{af}}$ skips failed picks and anchors on them; it is CoreSet's entry in the Holm family. The CoreSet column skips without anchoring ($\text{CoreSet}_{\text{xf}}$) and is bold by its own one-sided test. IF: Isolation Forest. The Random column is the library's random selector, which reuses one random stream every round at the pinned commit (Appendix W); the preregistered $\text{Random}_{\text{xf}}$ of Table 1 samples uniformly from uninspected items through $\text{CoreSet}_{\text{af}}$'s exploration step and uses none of that code. On Serengeti the library's random selector reaches a discovery AUC of 11.7, above $\text{Random}_{\text{xf}}$'s 7.25.

| | | Distance | | | | | | | Mixed | | | Density | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | Pool | CoreSet | $\text{CoreSet}_{\text{af}}$ | $k$NN | LOF | IF | NovelM. | MixedH. | MaxH. | NNDB | BADGE | TypiC. | ProbC. | $k$-m++ | Random |
| D1 iWildCam | NAIVE | 322.6 | — | 298.9 | 153.4 | 78.6 | 119.1 | 226.0 | 24.0 | 8.7 | 169.7 | 10.8 | 1.9 | 41.6 | 103.3 |
| | skip | **392.2** | **429.4** | **367.6** | **346.6** | **97.2** | **299.8** | **344.8** | **213.9** | **117.6** | 160.2 | **35.9** | **54.2** | 45.4 | 107.7 |
| D2 Serengeti | NAIVE | 42.5 | — | 48.8 | 64.3 | 6.9 | 5.6 | 14.6 | 0.0 | 0.0 | 18.1 | 0.1 | 0.0 | 4.2 | 9.9 |
| | skip | **53.6** | **71.2** | **63.5** | **86.2** | **13.8** | **53.5** | **54.6** | **6.3** | 2.0 | 18.3 | **3.9** | 0.0 | 2.1 | 11.7 |

of Appendix B beat the density-seeking ones in a random-effects meta-analysis. H3 says that on iWildCam and Serengeti CoreSet beats random under REMOVED and loses to it under NAIVE.

### 3.2 RESULTS

**H2 is confirmed: anchor on failed picks.** H2 is confirmed on both datasets (Table 1). Anchoring raises discovery AUC over skipping alone by +9.5% on iWildCam and +32.3% on Serengeti. $\text{CoreSet}_{\text{xf}}$ also beats $\text{Random}_{\text{xf}}$ by +227.9% and +849.1%, respectively. All four components reject after Holm (raw $p = 2^{-16}$ each). The non-decisional meta-analyses miss $\alpha$ ($p = 0.097$ for (b), 0.151 for (a)). All 8 secondary H2 components (final count and early AUC) also reject after Holm (Appendix S). In an exploratory rerun after the verdicts, CLIP ViT-L/14 embeddings give the same

three verdicts (Appendix Q). Excluding classes observed during initialization preserves all seedwise H2 AUC wins; the anchoring effects become $+9.3\%$ and $+35.0\%$ (Appendix A).

**Exploratory: comparing pool conventions.** Table 2 compares NAIVE and SELECTABLE. SELECTABLE beats NAIVE for 10 and 8 of 12 non-random selectors after Holm, including $\text{CoreSet}_{\text{af}}$ with anchoring. CoreSet gains at each step from NAIVE to skipping to anchoring, but initialization and stopping also change at the first step. Under skipping, the six best selectors on both datasets all seek distance, including MixedHalf. $\text{CoreSet}_{\text{af}}$ leads on iWildCam; on Serengeti LOF leads and $\text{CoreSet}_{\text{xf}}$ ranks fifth. Preregistered secondary comparisons favor $\text{CoreSet}_{\text{af}}$ over 14 of 15 arms on iWildCam and 13 of 15 on Serengeti after Holm (Table S.2). These comparisons carry no confirmatory weight. The exceptions are registry CoreSet, identical to $\text{CoreSet}_{\text{af}}$, and LOF on Serengeti, ahead in point estimate.

**H1 is not confirmed: distance is not uniformly better.** H1 is not confirmed. The contrast is large and positive on iWildCam, Serengeti and PlantNet but $-15.2\%$ on Nepenthaceae, and the pooled +122.0% has $p = 0.045$ with $I^2 = 100\%$. An exploratory analysis links the sign to where rare items sit. Their median $k$NN-distance percentile is 66.8 to 93.3 on the other datasets but 48.1 on Nepenthaceae. This association does not establish isolation as a necessary or sufficient condition for a distance advantage. The reversal also depends on the encoder: with PlantCLEF embeddings the Nepenthaceae contrast turns positive ($+7.9\%$, Appendix T). Seeking distance can also lose under skipping: on CIFAR-100-LT, $\text{Random}_{\text{xf}}$ beats $\text{CoreSet}_{\text{xf}}$ with CLIP embeddings but not with DINOv2 (Appendix Q).

**H3 is not confirmed: no collapse within ten rounds.** H3 is not confirmed. CoreSet leads random under REMOVED (+135.2%, +106.0%) and also under NAIVE (+213.4%, +541.5%), so the collapse seen on the drone pool (Appendix C.2) does not replicate. Re-proposal, which drove that collapse in the exploratory drone-pool runs, does recur: under NAIVE $94\%$ of CoreSet's round-10 failed picks on iWildCam are re-proposals. Random does not catch CoreSet within ten rounds, and longer horizons were not tested; on the drone pool it overtakes NAIVE CoreSet at round 4.

# 4 WHY FAILED PICKS MATTER

## 4.1 A COMPACT EXPLORATORY COMPARISON ON THE DRONE POOL

The drone pool contains 7,717 canopy images, of which experts labeled 3,280 to species. The other 4,437 were never labeled to species; no expert tried and failed to label them. Failed picks on this pool are therefore simulated. We compare three frozen encoders on the same images and labels, so encoder agreement is not independent replication. All drone-pool results are exploratory, including the choice of REMOVED as primary after seeing COVERED results (Appendices C and X).

When images without a label are kept out of selection, selectors that seek distance found rare classes faster than selectors that seek density. The pool convention changes CoreSet's gain over random on DINOv3 from $-27.23\%$ under NAIVE to $+39.11\%$ under COVERED and $+46.26\%$ under REMOVED (Table C.1). These conventions expose different information; COVERED and REMOVED require advance knowledge of which items are unlabelable. Skipping failures alone is deployable and raises CoreSet to $+50.7\%$ over skipping random (Appendix C.2). Our audit of 19 tools found no common failed-pick convention, and scikit-activeml's own CoreSet also re-proposes failed picks unless candidates are restricted (Appendices O.1 and O.2).

## 4.2 AVOIDING FAILURES IS NOT THE SAME AS FINDING RARE CLASSES

An exploratory learned labelability comparison uses a class-balanced logistic regression fit each round on observed successes and failures. One variant filters candidates below predicted labelability 0.5; another weights CoreSet distances by predicted labelability. The filter fills an undersized batch from the remaining candidates. Both use skip-only CoreSet until successes and failures have been observed. Neither significantly beats $\text{CoreSet}_{\text{af}}$ on any encoder after Holm correction. On DINOv3 and DINOv2, both discover fewer rare classes than $\text{CoreSet}_{\text{af}}$ (reverse one-sided tests reject after Holm): discovery AUC is 9.5 to 12.7% lower, despite 150 to 202 fewer failed picks per

Table 3: **Exploratory: MegaDetector improves discovery, but anchoring no longer leads on filtered iWildCam.** Means across 16 paired seeds. Rare is the final number of distinct rare classes discovered; Failed counts post-initial failed picks. Every arm additionally charges the same original initialization. The filter acts only on remaining candidates; evaluation retains the original rare-class sets.

| | | iWildCam | | | Serengeti | | |
|---|---|---|---|---|---|---|---|
| Selector | MDv5a | AUC | Rare | Failed | AUC | Rare | Failed |
| $\text{CoreSet}_{\text{af}}$ | No | 429.4 | 56.6 | 238.2 | 71.2 | 8.8 | 420.9 |
| $\text{CoreSet}_{\text{af}}$ | Yes | 444.9 | 57.6 | 67.7 | 77.1 | 9.2 | 219.8 |
| Skip-only | No | 392.2 | 53.5 | 773.0 | 53.6 | 6.4 | 723.5 |
| Skip-only | Yes | 455.5 | 58.2 | 206.9 | 73.4 | 8.9 | 394.2 |
| $\text{Random}_{\text{xf}}$ | No | 120.2 | 21.1 | 934.3 | 7.2 | 1.4 | 671.9 |
| $\text{Random}_{\text{xf}}$ | Yes | 202.1 | 33.4 | 123.6 | 28.6 | 5.1 | 240.6 |

run. On PlantCLEF, the differences from $\text{CoreSet}_{\text{af}}$ are $-0.6\%$ and $+1.6\%$ and are not significant (Appendix T). The filter learns from inspection feedback; it is not a pretrained empty-frame detector.

Anchoring lowers nearby priority without removing images, but it may also suppress a rare class near a failure, so the confirmed benefit does not establish a universal mechanism. A separated-cluster model gives one account of why distance can expose small classes and why isolated unlabelable groups absorb budget (Appendix G). Exploratory density ablations support that account on the drone pool, but do not test every geometry (Appendix C.4).

### 4.3 A PRETRAINED DETECTOR CHANGES THE COMPARISON

We compare the three camera-trap selectors with and without MegaDetector v5a at a fixed animal-confidence threshold of 0.1 (agentmorris, 2022). The exploratory design was frozen before detector inference. All arms inspect the same original initial pool, charging 201 picks on iWildCam and 88 on Serengeti before filtering remaining candidates. The original rare sets of 87 and 15 classes and the ten-round budgets stay fixed; removed rare classes remain missed (Appendix E).

Filtering raises mean discovery AUC for all three selectors on both pools (Table 3), but changes which CoreSet variant leads. On filtered iWildCam, anchoring trails skipping alone by 10.56 AUC units (95% paired interval $[-16.06, -5.63]$), despite fewer failed picks. On filtered Serengeti, its advantage remains positive at 3.75 units ($[1.75, 5.88]$). Thus the unfiltered H2 result does not establish that anchoring should follow a pretrained filter.

This gain has a cost: filtering removes a mean 12.9 rare images on iWildCam and 1.0 on Serengeti, making 6 of 87 and 1 of 15 rare classes unreachable in every seed. Inference over each 15,000-image pool takes 27.1 and 32.4 minutes on an H100 GPU partition (Table E.1). MegaDetector's training corpus includes Snapshot Serengeti; exact image overlap on either pool is unknown. This is a comparison with an available filter, not independent detector validation. It complements the drone-pool learned-filter experiment, which uses only inspection feedback and simulated failures.

### 4.4 MOVING FAILURE GROUPS TOWARD RARE-CLASS CENTERS

An exploratory synthetic study moves compact failure groups toward rare-class centers while holding labelable images, initial pools and budgets fixed. We froze the design before these runs. Across 16 independent pools, anchoring's mean advantage decreases from 58.56 to 15.56 discovery-AUC units (Table 4). The paired zero-minus-full difference is 43.00 (95% interval $[33.81, 52.75]$, exact one-sided $p = 2^{-16}$). The mean advantage never reverses. This is a center-interpolation intervention, not verified neighborhood mixing: every failure's ten nearest neighbors remain failures at every level. The experiment therefore shows attenuation in this construction, not a general boundary where anchoring becomes harmful (Appendix D).

After verifying their input hashes, we also summarize the existing camera-trap pools descriptively. Failures occupy 95.97% and 97.30% of failure items' ten-neighbor sets on iWildCam and Serengeti;

Table 4: **Exploratory: anchoring's advantage decreases as failure centers approach rare-class centers.** Mean paired discovery-AUC difference, anchored minus skip-only CoreSet, across 16 synthetic pools. Intervals are descriptive paired percentile bootstrap 95% intervals. The parameter interpolates centers; it does not measure neighborhood overlap.

| Center interpolation | Mean Δ AUC | Paired 95% interval |
|---|---|---|
| 0.00 | 58.56 | [52.31, 65.31] |
| 0.25 | 44.31 | [35.50, 52.94] |
| 0.50 | 35.50 | [27.94, 43.06] |
| 0.75 | 19.75 | [15.44, 24.38] |
| 1.00 | 15.56 | [8.94, 22.12] |

rare items occupy 0.206% and 0.016%. This neither validates the synthetic mechanism independently nor proves why anchoring helped.

## 5 Related work

**Low-budget and cold-start active learning.** Settles (2010) and Ren et al. (2021) survey acquisition functions. Uniform random is hard to beat (Munjal et al., 2022; Chen et al., 2024), and in the low-budget regime simple clustering-based selection is a strong baseline (Pourahmadi et al., 2021); TypiClust (Hacohen et al., 2022), ProbCover (Yehuda et al., 2022) and MaxHerding (Bae et al., 2024) were designed for this regime and for classifier accuracy. CoreSet (Sener & Savarese, 2018) and Bayesian coresets (Pinsler et al., 2019) target coverage. Evaluation frameworks (Lüth et al., 2023; Werner et al., 2024; Beck et al., 2021) show that rankings move with training setup, initial pool, pool subsampling (Ji et al., 2023) and seeds, and collective outliers can make selectors lose to random (Karamcheti et al., 2021). Foundation embeddings, now also trained for biology (Stevens et al., 2024), change which selector wins (Gupte et al., 2024; Zhu et al., 2025; Rauch et al., 2025); Zhang et al. (2026) target class imbalance with foundation-model priors and score balanced accuracy, and Mullapudi et al. (2021) learn rare-category classifiers on a tight budget. Most of these studies score accuracy and assume every pick returns a label; we score distinct rare classes found when some return none.

**Open-set and cold-start selection.** Proactive learning (Donmez & Carbonell, 2008) models oracles that may decline to answer, and open-set active learning (Du et al., 2021; Ning et al., 2022; Park et al., 2022; Safaei et al., 2024) treats queries outside the known classes as wasted budget and learns to avoid them from both known- and unknown-class queries; its implementations disagree on whether such a query stays in the unlabeled pool (Appendix O.2). Cold-start selection (Wang et al., 2022; Xie et al., 2023; Chen et al., 2024) picks a first labeled set from unlabeled features alone and assumes every pick can be labeled. A detector can filter most empty camera-trap frames (Beery et al., 2019); Section 4.3 tests this intervention separately from the confirmatory setting, where only inspected failures reveal unlabelability. A novel class must be reached at selection time, not flagged at recognition time as in open-set species recognition (Chen et al., 2025).

**Rare-category detection and active search.** Finding one example of each minority class with few queries is rare-category detection (Pelleg & Moore, 2004; He & Carbonell, 2007), and NNDB adapts He & Carbonell (2007). Active search (Garnett et al., 2012) seeks as many members of a rare positive class as possible under a query budget, and similarity search scales it to large pools (Coleman et al., 2022); our endpoint instead counts distinct rare classes. LOF (Breunig et al., 2000) and Isolation Forest (Liu et al., 2008) outlier scores rank by isolation, the signal distance-seeking selectors share. Discovery and classification place conflicting demands on the query criterion (Hospedales et al., 2013; Xie et al., 2025). Closest to our setting, Kurinchi-Vendhan & Beery (2026) compare selectors including CoreSet and BADGE on frozen DINOv3 and Perch embeddings over ecological datasets, score discovery time alongside accuracy, and find that rare examples in dense regions of other classes are found later, as in our H1 reversal (Section 3). We add mechanism ablations, pool conventions and separate tests of joint and individual claims. ADD (Kothawade et al., 2022) instead discovers initially unknown concepts using submodular conditional gain before targeting ac-

quired exemplars. Its discovery stage needs no rare-class query examples; Appendix A relates its facility-location objective to our MaxHerding baseline, not to the full ADD pipeline.

**Long-tail benchmarks and tooling.** The iNaturalist and LifeCLEF series (Van Horn et al., 2018; Joly et al., 2020; Goëau et al., 2024a) are canonical long-tail benchmarks, joined by moth-trap insects (Jain et al., 2024) and drone tree-species imagery (Cloutier et al., 2024), and active learning has aided camera-trap identification (Norouzzadeh et al., 2021). scikit-activeml, libact, small-text and BaaL (Herde et al., 2025; Yang et al., 2017; Schröder et al., 2023; Atighehchian et al., 2020) treat a failed pick differently: scikit-activeml and libact re-propose it, small-text removes it, and BaaL raises an error on a missing label when its dataset accepts real annotations (Appendix O.2). Lightly, FiftyOne and Labelbox (Lightly, 2026; Voxel51, 2026; Labelbox, 2026) rank or queue images; any of these can be wrapped in the `labelfirst Strategy` protocol. On analysis recording, see Forde & Paganini (2019), Cockburn et al. (2020) and Lipton & Steinhardt (2019).

## 6 Limitations and recommendations

**External validity.** The confirmed claim is H2, on two camera-trap datasets with one preregistered encoder. It concerns unfiltered pools, and in an exploratory check skipping CoreSet loses to skipping random on CIFAR-100-LT with CLIP embeddings (Appendix Q). The exploratory MegaDetector comparison reverses the anchoring advantage on iWildCam after filtering, while retaining it on Serengeti; training-corpus overlap and rare classes lost to filtering limit interpretation (Section 4.3). The drone-pool results come from one benchmark with three encoders, and their preregistered tests, H1 and H3, are not confirmed. On Nepenthaceae with DINOv3 the distance-versus-density contrast reverses ($-15.2\%$), though not with PlantCLEF embeddings. Exploratory audits on a PlantNet-300K subset and a second iNaturalist slice also favor distance-seeking selectors but lack complete input provenance (Appendix Q). The rare set comes from a labeled convenience sample, and repeated views of one tree are not grouped (Appendix H).

**Statistical scope.** Serengeti's $\text{Random}_{\text{xf}}$ arm is a low baseline: by round ten it finds $1.4$ of $15$ rare classes against $6.4$ for $\text{CoreSet}_{\text{xf}}$, with a mean discovery AUC of only 7. H2b's large percentage effect there partly reflects that small baseline. Seeds vary only the initial pool and selector randomness, so the $p$-values ignore the sampling of images or datasets (Bouthillier et al., 2021).

**Simulator.** The simulator assumes a fixed pool and a perfect oracle; Appendices W and X give the random-stream check, under which all three verdicts stay the same, and ten further disclosures.

**Accuracy is a separate endpoint.** In exploratory linear probes (covered test rows, Appendix Q), CoreSet changes mean test accuracy by $-1.4$ to $+0.4\%$ and class-balanced accuracy by $-0.4$ to $+3.3\%$ relative to random on the three drone-pool encoders. NovelMargin and gradient BADGE raise class-balanced accuracy by 7.7 to $12.5\%$.

**Reporting recommendations.** State the pool convention for unlabelable items; on the drone pool it decided the sign against random and the closest comparison. Report initialization and stopping too, as they also differ between conventions (Table T.1). Test joint claims by intersection-union and individual ones by Holm, and pair each non-rejection with an equivalence test and power.

## 7 Conclusion

In the tested unfiltered loops, keeping failed picks as distance anchors helps CoreSet find rare classes. On two camera-trap datasets with DINOv3 embeddings, anchoring beats skipping alone and CoreSet that skips failed picks beats random selection that skips them, confirming H2. With an exploratory MegaDetector filter, the anchoring advantage reverses on iWildCam and stays positive on Serengeti, and filtering raises discovery overall but loses some rare classes. Exploratory learned filters avoid more failures yet lower discovery AUC on two drone-pool encoders. H1 and H3 are not confirmed: the pooled distance-versus-density contrast misses $\alpha$, and under re-proposal CoreSet stays ahead of random within ten camera-trap rounds. The `labelfirst` library and the per-seed evidence let others rerun the audit on their own pools.

## AI USE STATEMENT

We used large language model assistants (Claude and Codex) to write and test code, run and check experiments, and edit the prose. The authors directed the work, checked every reported number against the committed evidence files, and take full responsibility for the content.

## ETHICS STATEMENT

`labelfirst` is a decision-support tool for annotation teams in long-tailed visual domains. This paper evaluates it on close-up drone imagery of a tropical forest canopy, where earlier discovery of rare classes can support biodiversity monitoring, and on public releases used as distributed under their licenses: iWildCam 2022 and Snapshot Serengeti camera-trap frames (CDLA-Permissive-1.0), PlantNet-300K (CC BY 4.0), an iNaturalist slice (per-photo licenses, recorded in the manifest) and RFCx audio. Our curated BCI pool is not yet publicly released; a separate labeled set of close-ups from the same site is public (Saha et al., 2026). We make no claim beyond these datasets. Camera traps can capture people: the Snapshot Serengeti release withholds its human-labeled images and our builder also drops sequences annotated as human; for iWildCam 2022 we used the released images as distributed. We redistribute no images from any source. Imagery of rare species can carry location information that exposes the species to collection or disturbance, so a team applying the tool to such data should control access to image coordinates and to the per-image outputs that link a rare class to a place. The tool aims to reduce annotation cost but does not replace expert annotators, and the audit models the annotator as a perfect oracle. The main risk is over-reliance on a verdict: a team that trusts a PASS without checking the intervals, the pool convention and the per-seed trajectories may deploy a selector whose advantage is real but too small to matter.

## REPRODUCIBILITY STATEMENT

**Environment.** The audit runs on Python $\geq$ 3.11 with `numpy`, `scipy`, `scikit-learn`, `threadpoolctl`, `pandas`, `typer` and `rich`. The `labelfirst` package ships the simulator, the selectors, the audit (Appendix C.1) and a provenance envelope (Appendix U). Each evidence file records the library version or commit that produced it: commit `058e4ab` (tag `v0.10.1-iclr`) for the REMOVED, baseline, mechanism and most sensitivity files; version 0.10.0 for the COVERED files; version 0.10.1 for the NAIVE, multi-dataset and stopping files; commit `575ac75` (version 0.10.2) for the confirmatory runs; commits `ec4733f` and `1195dba` for the labelability analyses; and commit `7bc7434` for the mixing-selector sweep. `v0.10.2-iclr` is the released pin and adds the renamed $k$-means++ entry point. Random-based arms at these commits reuse one random stream per round (Appendix W). Reproducing the TypiClust and gradient-BADGE trajectories requires scikit-learn 1.9.$x$; the other selectors reproduced exactly under 1.8.0 on DINOv3. The REMOVED, mechanism and sensitivity files record 1.9.0; the COVERED, NAIVE and multi-dataset files record no version. The bundled checks recompute every statistic from saved trajectories without a GPU. Re-running real-data selection requires the embeddings and labels, which are not bundled; the synthetic center-interpolation pools are bundled for full replay.

**Data access.** The anonymized supplementary archive bundles the codebase, the analysis scripts and the saved JSON evidence: per-seed, per-round trajectories for every selector and convention. The same code and evidence can be browsed at the anonymized repository `https://anonymous.4open.science/r/labelfirst-paper-2CC5/`. Every table and every number in the main text, captions included, is regenerated or checked by a script with a `--check` mode that exits nonzero on drift, except descriptions of external sources (flight height, encoder input resolutions and the PlantCLEF pretraining corpus): `audit_stats.py` and `make_main_tables.py` for the tables, `check_claims.py` for the prose and caption numbers and the two COVERED tables of Appendix I, `confirm_latex.py` for the confirmatory tables and macros, and `regen_evidence_envelope.py`, `pool_only_envelope.py`, `mechanism_check.py`, `sensitivity_check.py` and `rarefaction.py` for the envelopes behind them. `analyze.py` and `confirm_latex.py` in `scripts/confirm` recompute the confirmatory decisions and Table 1 from the committed runs, and `prereg/eligibility.json` records the SHA-256 hashes of each confirmatory manifest

and embedding file. The protocol was frozen in commit `367064b`, pushed, and amended before any confirmatory run was submitted (final `725da03`, Appendix X); it is not in a third-party registry, and the server-side push log and job-scheduler records are available to the area chair on request. The labeled BCI images of our pool are not yet publicly released; a separate labeled set of close-ups from the same site is public (Saha et al., 2026). The archive includes the per-seed trajectories and the SHA-256 hashes of the three drone-pool embedding files (about 405 MB in total) and of the labels, recorded in each evidence file, but not the images, embeddings or labels.

## REFERENCES


Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

agentmorris. MegaDetector v5a: checkpoint md_v5a.0.0.pt. Model release v5.0, 2022. URL https://github.com/agentmorris/MegaDetector/releases/tag/v5.0. Released June 15, 2022.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1027–1035, 2007.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://iclr.cc/virtual_2020/poster_ryghZJBKPS.html.

Parmida Atighehchian, Frédéric Branchaud-Charron, and Alexandre Lacoste. Bayesian active learning for production, a systematic study and a reusable library. *arXiv preprint arXiv:2006.09916*, 2020.

Wonho Bae, Junhyug Noh, and Danica J. Sutherland. Generalized coverage for more robust low-budget active learning. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.1007/978-3-031-73010-8_19.

Nathan Beck, Durga Sivasubramanian, Apurva Dani, Ganesh Ramakrishnan, and Rishabh Iyer. Effective evaluation of deep active learning on image classification tasks. *arXiv preprint arXiv:2106.15324*, 2021.

Sara Beery, Dan Morris, and Siyu Yang. Efficient pipeline for camera trap image review. *arXiv preprint arXiv:1907.06772*, 2019.

Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iWildCam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021.

Roger L. Berger. Multiparameter hypothesis testing and acceptance sampling. *Technometrics*, 24 (4):295–300, 1982. doi: 10.2307/1267823.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems (MLSys)*, 3, 2021.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 93–104, 2000. doi: 10.1145/342009.335388.

Anne Chao. Nonparametric estimation of the number of classes in a population. *Scandinavian Journal of Statistics*, 11:265–270, 1984.

Anne Chao and Lou Jost. Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size. *Ecology*, 93(12):2533–2547, 2012. doi: 10.1890/11-1952.1.

Liangyu Chen, Yutong Bai, Siyu Huang, Yongyi Lu, Bihan Wen, Alan Yuille, and Zongwei Zhou. Making your first choice: To address cold start problem in medical active learning. In *Medical Imaging with Deep Learning (MIDL)*, volume 227 of *Proceedings of Machine Learning Research*, pp. 496–525. PMLR, 2024. URL https://proceedings.mlr.press/v227/chen24a.html.

Yuyan Chen, Nico Lang, B. Christian Schmidt, Aditya Jain, Yves Basset, Sara Beery, Maxim Larrivée, and David Rolnick. Open-Insect: Benchmarking open-set recognition of novel species in biodiversity monitoring. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 38, 2025.

Myriam Cloutier, Mickaël Germain, and Etienne Laliberté. Influence of temperate forest autumn leaf phenology on segmentation of tree species from UAV imagery using deep learning. *Remote Sensing of Environment*, 311:114283, 2024. doi: 10.1016/j.rse.2024.114283.

Andy Cockburn, Pierre Dragicevic, Lonni Besançon, and Carl Gutwin. Threats of a replication crisis in empirical computer science. *Communications of the ACM*, 63(8):70–79, 2020. doi: 10.1145/3360311.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? Statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

Cody Coleman, Edward Chou, Julian Katz-Samuels, Sean Culatana, Peter Bailis, Alexander C. Berg, Robert Nowak, Roshan Sumbaly, Matei Zaharia, and I. Zeki Yalniz. Similarity search for efficient active learning and search of rare concepts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6402–6410, 2022.

Pinar Donmez and Jaime G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM)*, pp. 619–628, 2008. doi: 10.1145/1458082.1458165.

Pan Du, Suyun Zhao, Hui Chen, Shuwen Chai, Hong Chen, and Cuiping Li. Contrastive coding for active learning under class distribution mismatch. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8927–8936, 2021.

Bradley Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185, 1987. doi: 10.1080/01621459.1987.10478410.

Jessica Zosa Forde and Michela Paganini. The scientific method in the science of machine learning. *arXiv preprint arXiv:1904.10922*, 2019.

Camille Garcin, Alexis Joly, Pierre Bonnet, Jean-Christophe Lombardo, Antoine Affouard, Mathias Chouet, Maximilien Servajean, Titouan Lorieul, and Joseph Salmon. Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1 (NeurIPS Datasets and Benchmarks 2021)*, 2021.

Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard Mann. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012. URL https://arxiv.org/abs/1206.6406.

Hervé Goëau, Vincent Espitalier, Pierre Bonnet, and Alexis Joly. Overview of PlantCLEF 2024: Multi-species plant identification in vegetation plot images. In *Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum*, volume 3740 of *CEUR Workshop Proceedings*, pp. 1978–1988, 2024a.

Hervé Goëau, Jean-Christophe Lombardo, Antoine Affouard, Vincent Espitalier, Pierre Bonnet, and Alexis Joly. PlantCLEF 2024 pretrained models on the flora of the south western Europe based on a subset of Pl@ntNet collaborative images and a ViT base patch 14 dinoV2. Zenodo, 2024b.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi: 10.1016/0304-3975(85)90224-5.

Sanket Rajan Gupte, Josiah Aklilu, Jeffrey J. Nirschl, and Serena Yeung-Levy. Revisiting active learning in the era of vision foundation models. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://arxiv.org/abs/2401.14555.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022. URL https://proceedings.mlr.press/v162/hacohen22a.html.

Joachim Hartung and Guido Knapp. On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in Medicine*, 20(12):1771–1782, 2001. doi: 10.1002/sim.791.

Jingrui He and Jaime G. Carbonell. Nearest-neighbor-based active learning for rare category detection. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Marek Herde, Minh Tuan Pham, Daniel Kottke, Alexander Benz, Lukas Lührs, Pascal Mergard, Christoph Sandrock, Jiaying Cheng, Atal Roghman, Mehmet Müjde, Lukas Rauch, and Bernhard Sick. scikit-activeml: A comprehensive and user-friendly active learning library. *Preprints*, 2025. doi: 10.20944/preprints202507.0252.v1.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Timothy M. Hospedales, Shaogang Gong, and Tao Xiang. Finding rare classes: Active learning with generative and discriminative models. *IEEE Transactions on Knowledge and Data Engineering*, 25(2):374–386, 2013. doi: 10.1109/tkde.2011.231.

Stuart H. Hurlbert. The nonconcept of species diversity: A critique and alternative parameters. *Ecology*, 52(4):577–586, 1971. doi: 10.2307/1934145.

Aditya Jain, Fagner Cunha, Michael James Bunsen, Juan Sebastián Cañas, Léonard Pasi, Nathan Pinoy, Flemming Helsing, JoAnne Russo, Marc Botham, Michael Sabourin, Jonathan Fréchette, Alexandre Anctil, Yacksecari Lopez, Eduardo Navarro, Filonila Perez Pimentel, Ana Cecilia Zamora, José Alejandro Ramirez Silva, Jonathan Gagnon, Tom August, Kim Bjerge, Alba Gomez Segura, Marc Bélisle, Yves Basset, Kent P. McFarland, David Roy, Toke Thomas Høye, Maxim Larrivée, and David Rolnick. Insect identification in the wild: The AMI dataset. In *ECCV*, 2024. arXiv:2406.12452.

Yilin Ji, Daniel Kaestner, Oliver Wirth, and Christian Wressnegger. Randomness is the root of all evil: More reliable evaluation of deep active learning. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

Alexis Joly, Hervé Goëau, Stefan Kahl, Benjamin Deneu, Maximillien Servajean, Elijah Cole, Lukáš Picek, Rafael Ruiz de Castañeda, Isabelle Bolon, Andrew Durso, Titouan Lorieul, Christophe Botella, Hervé Glotin, Julien Champ, Ivan Eggel, Willem-Pier Vellinga, Pierre Bonnet, and Henning Müller. Overview of LifeCLEF 2020: A system-oriented evaluation of automated species identification and species distribution prediction. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction (CLEF 2020)*, Lecture Notes in Computer Science, pp. 342–363. Springer, 2020. doi: 10.1007/978-3-030-58219-7_23.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher D. Manning. Mind your outliers! Investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7265–7281, 2021. doi: 10.18653/v1/2021.acl-long.564.

Suraj Kothawade, Nathan Beck, Krishnateja Killamsetty, and Rishabh Iyer. SIMILAR: Submodular information measures based active learning in realistic scenarios. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Suraj Kothawade, Shivang Chopra, Saikat Ghosh, and Rishabh Iyer. Active data discovery: Mining unknown data using submodular information measures. *arXiv preprint arXiv:2206.08566*, 2022. doi: 10.48550/arXiv.2206.08566. URL https://arxiv.org/abs/2206.08566.

Rupa Kurinchi-Vendhan and Sara Beery. Finding needles in the haystack: Transductive active labeling in ecology. *arXiv preprint arXiv:2606.03821*, 2026.

Labelbox. Labelbox documentation: send a batch to a labeling project. https://docs.labelbox.com/docs/batches, 2026. Accessed September 18, 2026.

Etienne Laliberté, Antoine Caron-Guay, Vincent Le Falher, Guillaume Tougas, Helene C. Muller-Landau, Gonzalo Rivas-Torres, Thomas R. Walla, Hugo Baudchon, Mélvin Hernandez, Adrian Buenaño, Anna Weber, Jeffrey Q. Chambers, Jomber Chota Inuma, Fernando Araúz, Jorge Valdes, Andrés Hernández, David Brassfield, Paulo Araujo Filho, Vicente Vasquez, Adriana Simonetti, Daniel Magnabosco Marra, Caroline de Moura Vasconcelos, Jarol Fernando Vaca, Geovanny Rivadeneyra, José Illanes, Luis A. Salagaje-Muela, and Jefferson Gualinga. Seeing the forest and the trees: a workflow for automatic acquisition of ultra-high resolution drone photos of tropical forest canopies to support botanical and ecological studies, 2025. bioRxiv preprint.

Egbert Giles Leigh, Jr. *Tropical forest ecology: a view from Barro Colorado Island*. Oxford University Press, 1999. doi: 10.1093/oso/9780195096026.001.0001.

Lightly. Lightly documentation: Examples and use cases. https://docs.lightly.ai/docs/selection-examples-and-use-cases, 2026. Accessed September 18, 2026.

Zachary C. Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship. *ACM Queue*, 17(1):45–77, 2019. doi: 10.1145/3317287.3328534.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Eighth IEEE International Conference on Data Mining (ICDM)*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Carsten T. Lüth, Till J. Bungert, Lukas Klein, and Paul F. Jaeger. Navigating the pitfalls of active learning evaluation: A systematic framework for meaningful performance assessment. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Shijie Ma, Fei Zhu, Zhun Zhong, Xu-Yao Zhang, and Cheng-Lin Liu. Active generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16890–16900, 2024. doi: 10.1109/cvpr52733.2024.01598.

Ravi Teja Mullapudi, Fait Poms, William R. Mark, Deva Ramanan, and Kayvon Fatahalian. Learning rare category classifiers on a tight labeling budget. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8403–8412, 2021. doi: 10.1109/ICCV48922.2021.00831.

Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. Towards robust and reproducible active learning using neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi: 10.1109/cvpr52688.2022.00032.

Kun-Peng Ning, Xun Zhao, Yu Li, and Sheng-Jun Huang. Active learning for open-set annotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41–49, 2022.

Mohammad Sadegh Norouzzadeh, Dan Morris, Sara Beery, Neel Joshi, Nebojsa Jojic, and Jeff Clune. A deep active learning system for species identification and counting in camera trap images. *Methods in Ecology and Evolution*, 12(1):150–161, 2021. doi: 10.1111/2041-210X.13504.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://arxiv.org/abs/2304.07193.

Dongmin Park, Yooju Shin, Jihwan Bang, Youngjun Lee, Hwanjun Song, and Jae-Gil Lee. Meta-Query-Net: Resolving purity-informativeness dilemma in open-set active learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Dan Pelleg and Andrew W. Moore. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems*, volume 17, 2004.

Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*, 2019.

E. J. G. Pitman. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130, 1937. doi: 10.2307/2984124.

Laura J. Pollock, Justin Kitzes, Sara Beery, Kaitlyn M. Gaynor, Marta A. Jarzyna, Oisin Mac Aodha, Bernd Meyer, David Rolnick, Graham W. Taylor, Devis Tuia, and Tanya Berger-Wolf. Harnessing artificial intelligence to fill global shortfalls in biodiversity knowledge. *Nature Reviews Biodiversity*, 1(3):166–182, 2025. doi: 10.1038/s44358-025-00022-3.

Kossar Pourahmadi, Parsa Nooralinejad, and Hamed Pirsiavash. A simple baseline for low-budget active learning. *arXiv preprint arXiv:2110.12033*, 2021.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000. doi: 10.1145/342009.335437.

Lukas Rauch, Moritz Wirth, Denis Huseljic, Marek Herde, Bernhard Sick, and Matthias Aßenmacher. No free lunch in active learning: LLM embedding quality dictates query strategy success. *arXiv preprint arXiv:2506.01992*, 2025.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021. doi: 10.1145/3472291.

Christian Röver, Guido Knapp, and Tim Friede. Hartung-Knapp-Sidik-Jonkman approach and its modification for random-effects meta-analysis with few studies. *BMC Medical Research Methodology*, 15:99, 2015. doi: 10.1186/s12874-015-0091-1.

Bardia Safaei, Vibashan VS, Celso M. de Melo, and Vishal M. Patel. Entropic open-set active learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

Sulagna Saha, Arthur Ouaknine, Etienne Laliberté, Carol Altimas, Evan M. Gora, Adriane Esquivel Muelbert, Ian R. McGregor, Cesar Gutierrez, Vanessa E. Rubio, and David Rolnick. Understanding representation gaps across scales in tropical tree species classification from drone imagery. ICLR 2026 Workshop on Machine Learning for Remote Sensing, 2026. arXiv:2604.23019.

Christopher Schröder, Lydia Müller, Andreas Niekler, and Martin Potthast. Small-Text: Active learning for text classification in Python. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL)*, pp. 84–95, 2023. doi: 10.18653/v1/2023.eacl-demo.11.

Donald J. Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, 1987. doi: 10.1007/BF01068419.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2010. URL https://burrsettles.com/pub/settles.activelearning.pdf.

Kurex Sidik and Jeffrey N. Jonkman. A simple confidence interval for meta-analysis. *Statistics in Medicine*, 21(21):3153–3159, 2002. doi: 10.1002/sim.1262.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research (TMLR)*, 2026. URL https://openreview.net/forum?id=2NlGyqNjns.

Samuel Stevens, Jiaman Wu, Matthew J. Thompson, Elizabeth G. Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila M. Dahdul, Charles Stewart, Tanya Berger-Wolf, Wei-Lun Chao, and Yu Su. BioCLIP: A vision foundation model for the tree of life. In *CVPR*, 2024. arXiv:2311.18803.

Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific Data*, 2:150026, 2015. doi: 10.1038/sdata.2015.26.

Hans ter Steege, Nigel C. A. Pitman, Daniel Sabatier, et al. Hyperdominance in the Amazonian tree flora. *Science*, 342(6156):1243092, 2013. doi: 10.1126/science.1243092.

Devis Tuia, Benjamin Kellenberger, Sara Beery, Blair R. Costelloe, Silvia Zuffi, Benjamin Risse, Alexander Mathis, Mackenzie W. Mathis, Frank van Langevelde, Tilo Burghardt, Roland Kays, Holger Klinck, Martin Wikelski, Iain D. Couzin, Grant van Horn, Margaret C. Crofoot, Charles V. Stewart, and Tanya Berger-Wolf. Perspectives in machine learning for wildlife conservation. *Nature Communications*, 13:792, 2022. doi: 10.1038/s41467-022-27980-y.

Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. doi: 10.1109/cvpr.2018.00914.

Wolfgang Viechtbauer. Bias and efficiency of meta-analytic variance estimators in the random-effects model. *Journal of Educational and Behavioral Statistics*, 30(3):261–293, 2005. doi: 10.3102/10769986030003261.

Voxel51. FiftyOne brain documentation. https://docs.voxel51.com/brain/index.html, 2026. Accessed September 18, 2026.

Xudong Wang, Long Lian, and Stella X. Yu. Unsupervised selective labeling for more effective semi-supervised learning. In *European Conference on Computer Vision (ECCV)*, pp. 427–445, 2022. doi: 10.1007/978-3-031-20056-4_25.

Thorben Werner, Johannes Burchert, Maximilian Stubbemann, and Lars Schmidt-Thieme. A cross-domain benchmark for active learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Tian Xie, Jifan Zhang, Haoyue Bai, and Robert D. Nowak. Deep active learning in the open world. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=HkmymFPODz.

Yichen Xie, Han Lu, Junchi Yan, Xiaokang Yang, Masayoshi Tomizuka, and Wei Zhan. Active finetuning: Exploiting annotation budget in the pretraining-finetuning paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23715–23724, 2023.

Yao-Yuan Yang, Shao-Chuan Lee, Yu-An Chung, Tung-En Wu, Si-An Chen, and Hsuan-Tien Lin. libact: Pool-based active learning in Python. *arXiv preprint arXiv:1710.00379*, 2017.

Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. In *Advances in Neural Information Processing Systems*, 2022. doi: 10.52202/068431-1624.

Jiancheng Zhang, Meiqing Li, Qi Zhang, and Yinglun Zhu. Active learning with foundation model priors: Efficient learning under class imbalance. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. arXiv:2606.07630.

Ning Zhu, Xiaochuan Ma, Shaoting Zhang, and Guotai Wang. MedCAL-Bench: A comprehensive benchmark on cold-start active learning with foundation models for medical image analysis. *arXiv preprint arXiv:2508.03441*, 2025.

**Guide to the appendix.** The appendix has four groups. *Theory and synthetic data*: Appendices F and G give the synthetic analyses and the geometric model. *Drone pool and other exploratory datasets*: Appendices H to R cover these datasets. They give the reference dataset, COVERED, sensitivity, the H1 contrast, the drone-pool tests and full comparison, specification and power, per-encoder values, failed picks in real tools, embedding geometry, additional datasets and stopping. *Confirmatory study*: Appendices S and T give its secondary endpoints and exploratory analyses of its runs. *Reproducibility and provenance*: Appendices U to X give the reproduction command, the status of every analysis, random-number generation, and the analysis history and disclosures.

## A EXPLORATORY ANCHOR-RETENTION FOLLOW-UP

**New classes rather than reacquisitions.** Using the original saved H2 selections, we reconstruct three endpoints: the registered acquisition-round count; only rare classes absent from the initial pool; and the union of initial and acquired rare classes. The original rare set, inspections and ten checkpoints remain fixed. The original endpoint reproduces exactly. Removing reacquisitions preserves every seedwise H2 AUC win on both datasets. The total-campaign and novel-only absolute paired differences coincide because initialization is shared. These post hoc sensitivities do not change the registered verdicts. Intervals in Table A.1 are descriptive paired percentile bootstrap intervals conditional on the fixed pools. The sign-flip test assumes null sign exchangeability; the all-positive AUC contrasts also give the same one-sided sign-test probability for a win-probability null, which is a different estimand from mean log advantage.

Table A.1: **Exploratory endpoint sensitivity on the original H2a selections.** Mean discovery AUC, anchored CoreSet ($\text{CoreSet}_{\text{af}}$) versus skipping alone. Novel-only excludes classes already present at initialization; total campaign includes them at every acquisition-round checkpoint. Shared initialization makes their absolute paired contrasts identical.

| Dataset | Endpoint | $\text{CoreSet}_{\text{af}}$ | Skip-only | Paired difference [95% CI] |
|---|---|---|---|---|
| iWildCam | Registered | 429.4 | 392.2 | 37.3 [33.3, 41.1] |
| iWildCam | Novel-only | 425.6 | 389.3 | 36.3 [32.4, 40.1] |
| iWildCam | Total campaign | 445.6 | 409.3 | 36.3 [32.4, 40.1] |
| Serengeti | Registered | 71.2 | 53.6 | 17.6 [14.9, 20.1] |
| Serengeti | Novel-only | 69.3 | 51.2 | 18.1 [16.0, 20.3] |
| Serengeti | Total campaign | 73.1 | 54.9 | 18.1 [16.0, 20.3] |

**Feedback cadence at fixed cost.** We hold the original initial inspections and total acquisition budget fixed, partitioning that budget into one, five, ten or twenty nearly equal batches. Both selectors temporarily anchor every within-batch pick; their difference is whether failed anchors survive between batches. Fully anchored CoreSet's acquisition order is identical across all four schedules on every tested seed and filtering condition, while skip-only changes. To compare schedules, we score the cumulative novel-class count after each ordered inspection and divide its sum by the common acquisition budget. This retrospective ordering metric is not the registered ten-checkpoint AUC, nor

does it assume feedback arrives before a batch completes. On unfiltered iWildCam, the anchoring advantage at ten batches has an interval above zero, while the single-batch interval includes zero (Table A.2). The two settings still differ in their treatment of initial failures. On filtered iWildCam, none of the schedules establishes a positive advantage. The choice to retain failure anchors therefore interacts with feedback cadence; the anchoring advantage on this exploratory ordering metric depends on batching.

Table A.2: **Exploratory fixed-budget cadence sensitivity.** Anchored minus skip-only mean novel-class coverage along the ordered acquisition sequence, with descriptive seed-paired percentile bootstrap intervals. Initialization and total inspection budget are shared across schedules. This metric differs from ten-round discovery AUC.

| Dataset | Batches | Unfiltered $\Delta$ [95% CI] | Filtered $\Delta$ [95% CI] |
|---|---|---|---|
| iWildCam | 1 | $-0.22$ $[-0.60, 0.14]$ | $-1.14$ $[-1.60, -0.73]$ |
| iWildCam | 5 | 1.91 [1.52, 2.24] | $-0.89$ $[-1.41, -0.45]$ |
| iWildCam | 10 | 3.98 [3.59, 4.35] | $-0.62$ $[-1.15, -0.16]$ |
| iWildCam | 20 | 6.60 [6.12, 7.04] | $-0.07$ $[-0.59, 0.35]$ |
| Serengeti | 1 | 0.21 [0.03, 0.41] | $-0.08$ $[-0.33, 0.18]$ |
| Serengeti | 5 | 0.93 [0.71, 1.14] | 0.35 [0.16, 0.53] |
| Serengeti | 10 | 1.73 [1.54, 1.92] | 0.45 [0.25, 0.66] |
| Serengeti | 20 | 2.34 [2.13, 2.55] | 0.79 [0.60, 0.97] |

**Conditional-gain comparison.** ADD's unit-strength facility-location conditional gain (Kothawade et al., 2022) has the same greedy marginal as MaxHerding when the kernel, coverage support and conditioning set match. We verify this identity algebraically and against the pinned implementation, allowing numerical tie differences. Our existing MaxHerding uses a Gaussian kernel and, under SELECTABLE, conditions on every inspected image. Rescoring its original selections for novel-only discovery leaves it below both CoreSet variants on every seed of both camera-trap pools. This is not a reproduction of full ADD: learned features, conditioning on known concepts, nonunit conditioning strength and subsequent targeting are not evaluated.

**What static suppression does not explain.** At each saved batch-start history, we compare distances to successful inspections with distances to all inspections, keeping the history and eligible candidates fixed. Failure anchors lower rare-candidate scores, but may lower competing candidates' scores more. Static top-$k$ novel-class coverage increases in every dataset, history-arm and filtering panel, including filtered iWildCam. This diagnostic therefore does not explain that dataset's negative anchoring effect. Static scores omit the temporary anchors added during greedy batch selection and cannot establish counterfactual discovery trajectories or a deployment rule. Uninspected labels enter only this retrospective evaluation.

## B SELECTORS

We group selectors by what their score rewards in a candidate. A *distance-seeking* score grows with the candidate's distance from the labeled set or from its own neighbors. A *density-seeking* score grows with the number or closeness of its neighbors. A *mixed* score rewards both.

**Distance-seeking.** *CoreSet* (Sener & Savarese, 2018) greedily picks the unlabeled point whose minimum cosine distance to the labeled set is largest, adding each pick to the reference set before the next (farthest-first traversal, Gonzalez, 1985). *kNN distance* (Ramaswamy et al., 2000) scores each unlabeled point by its cosine distance to its 10th nearest neighbor within the unlabeled pool. *LOF* (Breunig et al., 2000) (20 neighbors) and *Isolation Forest* (Liu et al., 2008) (scikit-learn defaults, Pedregosa et al., 2011) are refit on the unlabeled pool each round and pick the most outlying points. LOF compares a point's local density with its neighbors', so it favors points sparser than their surroundings. *NovelMargin*, loosely based on active generalized category discovery (Ma et al., 2024), is label-aware: it scores points by distance to the labeled class prototypes, up-weights points between two prototypes, and spreads its picks with $k$-center suppression. *MixedHalf* draws half its batch from CoreSet, half at random.

**Mixed.** *MaxHerding* (Bae et al., 2024) greedily maximizes kernel coverage, whose gain rewards a candidate with many uncovered neighbors far from the points already chosen. *NNDB* (He & Carbonell, 2007), adapted to batches, scores the largest drop in local density from a point to its neighbors. Like the multi-class NNDM of the same paper, it masks the neighborhood of every labeled point, so it seeks the dense edge of a compact cluster far from labels. *BADGE* (Ash et al., 2020) fits a logistic head on the labels so far and runs $k$-means++ on last-layer gradient embeddings, whose norm grows with the head's uncertainty.

**Density-seeking.** *TypiClust* (Hacohen et al., 2022) follows the reference code: it clusters all rows into $\min(|\text{labeled}| + k, 500)$ groups, drops groups of five or fewer, and takes the most typical point of the least-labeled groups first. *ProbCover* (Yehuda et al., 2022) greedily maximizes coverage by cosine balls of radius $\delta$. The paper's purity rule sets $\delta$ by $k$-means with one cluster per ground-truth class, which gives $\delta = 0.10$ on DINOv3 and DINOv2 and $0.08$ on PlantCLEF. That rule uses the class count, a label leak we remove in Appendix C.4. *$k$-means++* seeding (Arthur & Vassilvitskii, 2007) runs on the features of the unlabeled pool and ignores the labeled set. scikit-learn's greedy variant samples several candidates in proportion to squared distance and keeps the one that most reduces inertia, which favors dense regions. It is a feature-space surrogate, not BADGE.

We exclude submodular mutual-information selectors such as GCMI and FLQMI (Kothawade et al., 2021) because they need a query set of rare examples, which would leak the target labels into a cold-start selector. This exclusion does not apply to seed-free submodular conditional gain, as used by ADD (Kothawade et al., 2022).

# C Exploration on one drone pool

**Status.** Every result in this section is exploratory, including the choice of REMOVED as primary after the COVERED results were seen (Appendices V and X). Values are for DINOv3; Table N.1 gives the others.

## C.1 Pool, encoders and tests

The drone pool is a curated set of 7,717 close-up drone images of the Barro Colorado Island (BCI) forest canopy in Panama (Leigh, 1999), photographed from 6 m above selected crowns with the workflow of Laliberté et al. (2025). Of these, 3,280 were labeled to species by expert botanists, covering 252 distinct species, 157 of them rare (at most 5 ground-truth examples). The other 4,437 images were never labeled to species, so each is an unlabelable item. No expert tried and failed to label them, so the drone-pool failed picks are simulated. Nothing in the protocol is biological: a rare class is any under-represented label. We embed every image with three frozen encoders: DINOv3 ViT-L/16 (Siméoni et al., 2026) (1,024 dimensions, 224 px), DINOv2-Giant (Oquab et al., 2024) (1,536, 224 px), and PlantCLEF2024 (Goëau et al., 2024a;b), a DINOv2 ViT-B/14 fine-tuned on 1.4 million plant images (768, 672 px; checkpoint in Appendix U). The three encoders see the same images and labels, so agreement across them shows consistency within one pool, not replication. Runs use $K = 10$ rounds, $k = 200$, an initial pool of 200, and seeds 0 to 15, identical across selectors. The random-based selectors reuse one random stream per round (Appendix W). The full budget, $200 + 10 \times 200 = 2{,}200$ labels, is 67% of the 3,280 labeled images, so late rounds draw from a depleted pool.

**Tests.** Gains between the leading selectors are small, so power and the number of seeds decide which claims hold (Colas et al., 2018; Agarwal et al., 2021; Werner et al., 2024). For each comparison of CoreSet with an alternative $\pi^{(j)}$, we compute the 16 paired differences in the endpoint. A one-sided exact sign-flip permutation test (Pitman, 1937) on their mean enumerates all $2^{16}$ sign vectors, so the smallest attainable $p$ is $2^{-16} \approx 1.5 \times 10^{-5}$ (Appendix M.1). The relative gain $100\% \cdot (\bar{a}^{\star}/\bar{a}^{(j)} - 1)$ carries a paired percentile bootstrap 95% interval from 10,000 resamples. All tests are one-sided at $\alpha = 0.025$. The *joint* claim that CoreSet beats every member of a family holds when every component rejects (intersection-union, Berger, 1982); which members it beats is a separate family, tested with Holm (Holm, 1979). The *core* family has six alternatives: MixedHalf, MaxHerding, NovelMargin, ProbCover, TypiClust and $k$-means++. Appendix L gives the extended family, the equivalence tests and power.

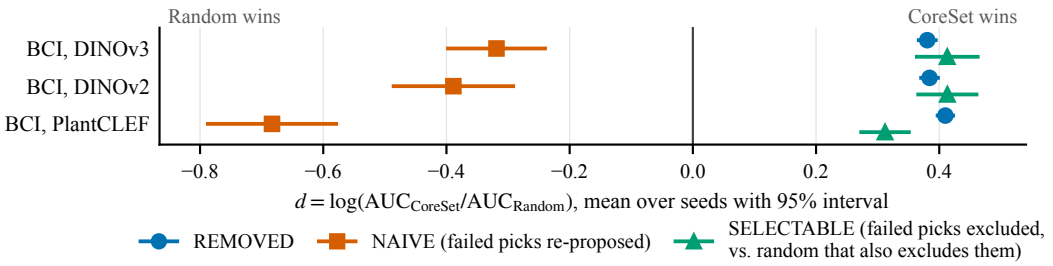



Figure C.1: **On the drone pool, the pool convention flips CoreSet against random; the preregistered test did not replicate the flip.** Paired log ratio $d = \log(\mathrm{AUC}_{\mathrm{CoreSet}}/\mathrm{AUC}_{\mathrm{Random}})$ of discovery AUC per seed, mean over 16 seeds with a Student-$t$ 95% interval; $d > 0$ means CoreSet finds rare classes faster. REMOVED drops the unlabelable items before selection. Under NAIVE they stay in the pool, and the loop re-proposes a failed pick every round. SELECTABLE skips failed picks for both selectors (Table U.1). All points are exploratory.

### C.2 THE POOL CONVENTION CHANGES THE RANKING

On this pool, the pool convention changes both the sign of CoreSet's gain over random and its closest comparison (Figure C.1 and Table C.1). Under NAIVE, a failed pick leaves the labeled set unchanged, so the loop re-proposes it in every later round. The unlabelable items sit farther out than most labeled images (Appendix T), so CoreSet, which seeks distance, spends its budget on them again and again. CoreSet falls 27.2% below random on DINOv3. On this pool the collapse tracks re-proposal rather than CoreSet itself: $k$NN distance falls 27.9% below random and NovelMargin 20.2%.

Skipping failed picks (SELECTABLE) removes the collapse with no labelability model. It lifts CoreSet's mean discovery AUC to +71.2% over NAIVE random and +50.7% over a random selector that also skips them. CoreSet leads on all 16 seeds, with exact $p \leq 1.6 \times 10^{-5}$ against both random arms on every encoder. Skipping still pays once for each unlabelable item it picks, so SELECTABLE CoreSet stays 46.9% below CoreSet under REMOVED on DINOv3. COVERED marks every unlabelable item as an anchor from the start. It avoids that loss but leaves CoreSet and NovelMargin within 1% (Appendix I). COVERED is not neutral either: a cross-validated logistic probe on the frozen embedding separates unlabelable from labeled images with ROC AUC 0.71 on DINOv3 (Appendix H). Its anchors thus sit in regions whose labelability differs from the rest, and they shift every geometry-based selector in a direction we cannot predict. REMOVED drops the unlabelable items instead, so every candidate is labelable.

**Re-proposal in a real library.** We ran scikit-activeml v1.0.0's own CoreSet cold-start on the same pool, rare classes and endpoint. With the library's default candidates, it re-proposes 197 to 199 of its 200 picks per round by round 10. It falls 19.7% below the library's random sampler on DINOv3, ahead on at most 1 of 16 seeds on any encoder. Passing only uninspected images as candidates puts it +46.0% above random on DINOv3 and on all 16 seeds of every encoder ($p = 2^{-16}$), within 2.1% of our SELECTABLE CoreSet's mean discovery AUC (Appendix O.1). Tools share no convention. We audited 19 libraries, research repositories and one annotation tool at pinned commits (Appendix O.2). Of these, 4 re-propose a failed pick, 2 remove it, 3 move it to a known out-of-distribution set, 1 forces a label, and the rest cannot produce one or leave it to the caller. Even two open-set methods from one lineage disagree: LfOSA leaves unknown-class picks in the unlabeled pool, and EOAL quarantines them. A ranking can therefore depend silently on the library that runs it.

### C.3 DISTANCE LEADS DENSITY WHEN EVERY CANDIDATE IS LABELABLE

Under REMOVED, CoreSet raises discovery AUC by +46.26% over random on DINOv3 (1121 against 766). It beats every member of the core family on all three encoders, with all six Holm rejections (Table L.1). Its closest competitor in the core family is NovelMargin, which CoreSet beats by +5.49%, with exact $p \leq 3.1 \times 10^{-5}$ and an interval above zero on every encoder. TOST

Table C.1: **Three pool conventions for the** $4{,}437$ **unlabelable items**, discovery AUC, $n_s = 16$. Gains in percent with $95\%$ percentile bootstrap intervals; exact one-sided sign-flip $p$ for CoreSet vs. NovelMargin. Table M.2 adds TOST and the joint claim.

| Convention | Encoder | CoreSet vs. random | CoreSet vs. NovelMargin | exact $p$ |
|---|---|---|---|---|
| NAIVE | DINOv3 | $-27.23\,[-32.36, -21.63]$ | $-8.83\,[-14.41, -3.24]$ | 0.995 |
| | DINOv2 | $-32.51\,[-37.85, -26.03]$ | $-11.31\,[-15.00, -7.36]$ | 1.000 |
| | PlantCLEF | $-49.51\,[-54.13, -44.68]$ | $-15.57\,[-20.20, -11.22]$ | 1.000 |
| COVERED | DINOv3 | $+39.11\,[+37.00, +41.32]$ | $+0.51\,[-0.28, +1.24]$ | 0.118 |
| | DINOv2 | $+40.54\,[+38.44, +42.73]$ | $-0.83\,[-1.51, -0.15]$ | 0.982 |
| | PlantCLEF | $+45.61\,[+43.53, +47.81]$ | $-0.53\,[-1.03, +0.01]$ | 0.966 |
| REMOVED | DINOv3 | $+46.26\,[+44.16, +48.54]$ | $+5.49\,[+4.60, +6.32]$ | $2^{-16}$ |
| | DINOv2 | $+46.81\,[+44.68, +48.98]$ | $+3.12\,[+2.39, +3.86]$ | $2^{-16}$ |
| | PlantCLEF | $+50.60\,[+48.55, +52.78]$ | $+2.19\,[+1.63, +2.75]$ | $2^{-15}$ |

does not declare the two equivalent on any encoder. Under COVERED this separation disappears (Appendix C.2).

Distance-seeking selectors share the lead: $k$NN distance is the strongest alternative on every encoder and ties CoreSet on DINOv2 (Appendix L). The contrast H1 tests also favors distance here (Appendix K). Random tracks its analytic rarefaction expectation (Hurlbert, 1971) to within $0.84\%$ of discovery AUC, and the density-seeking selectors sit on or near that curve (Figure J.1).

### C.4 MECHANISM: REMOVING THE DENSITY TERMS

Weakening the density terms closes much of the gap between the density-seeking selectors and CoreSet. TypiClust has two density components: a filter that drops clusters of five or fewer points, and a preference for the most typical point of each cluster. On DINOv3, removing the filter moves TypiClust from $-14.09\%$ to $+10.57\%$ against random. Also picking the *least* typical point moves it to $+30.02\%$. ProbCover at its purity-rule radius gains only $+0.60\%$ over random. With a label-free radius, the median nearest-neighbor distance ($\delta = 0.16$ on DINOv3), it gains $+19.27\%$. Larger balls plausibly cover the dense core in fewer picks and push the greedy selection out to sparse regions sooner. The ablations recover much, but not all, of the gap to CoreSet. Appendix G gives one account of the pattern in a separated-cluster model (Propositions 1 to 3). Farthest-first discovers a new class with every pick, whatever the class mass. TypiClust's size filter and typicality pick and ProbCover's greedy gain defer small classes. Unlabelable items far from the classes absorb farthest-first's budget.

**Three hypotheses.** The exploration yields three hypotheses, which we preregistered and tested (Section 3). H1: under REMOVED, distance-seeking selectors beat density-seeking ones. H2: where many items are unlabelable, skipping failed picks lets CoreSet beat random, and anchoring on them adds more. H3: CoreSet beats random under REMOVED, not under NAIVE.

## D EXPLORATORY CENTER-INTERPOLATION STUDY

**Frozen construction.** We fixed a machine-readable design before running this study. The earlier scratch configuration was unavailable, so we reused the existing synthetic generator with explicit settings chosen before outcomes: $1{,}249$ labelable items, $82$ classes, Zipf exponent $1.5$, $128$ dimensions and Gaussian noise scale $0.05$. This yields $54$ rare classes at the threshold of five. Seeds $0$ to $15$ generate independent labelable geometries with the same class-size distribution. Each pool adds $20$ failure groups of $30$ items. Each group independently draws a rare class uniformly with replacement and an isolated center from a normalized Gaussian vector. At interpolation values $0$, $0.25$, $0.5$, $0.75$ and $1$, its center follows the normalized linear interpolation toward the paired class's normalized empirical centroid. We add the same Gaussian noise, scale $0.02$, across all five condi-

Table D.1: **Exploratory synthetic results for every arm.** Means across 16 independent pools at each center-interpolation level. Failed picks count the five selection rounds; every run additionally charges the same 30 initial inspections within seed. All arms use the fixed set of 54 rare classes.

| Center interpolation | Selector | Discovery AUC | Final discoveries | Failed picks |
|---|---|---|---|---|
| 0.00 | $\text{CoreSet}_{\text{af}}$ | 159.88 | 51.50 | 11.69 |
| | $\text{CoreSet}_{\text{xf}}$ | 101.31 | 31.69 | 50.38 |
| | $\text{Random}_{\text{xf}}$ | 26.31 | 8.88 | 31.75 |
| | LOF | 130.00 | 34.56 | 0.00 |
| | Learned filter | 156.19 | 47.94 | 13.00 |
| 0.25 | $\text{CoreSet}_{\text{af}}$ | 159.56 | 51.50 | 11.69 |
| | $\text{CoreSet}_{\text{xf}}$ | 115.25 | 38.25 | 41.62 |
| | $\text{Random}_{\text{xf}}$ | 26.31 | 8.88 | 31.75 |
| | LOF | 133.69 | 36.00 | 0.00 |
| | Learned filter | 153.81 | 45.56 | 14.25 |
| 0.50 | $\text{CoreSet}_{\text{af}}$ | 162.56 | 51.44 | 11.69 |
| | $\text{CoreSet}_{\text{xf}}$ | 127.06 | 43.06 | 36.00 |
| | $\text{Random}_{\text{xf}}$ | 26.31 | 8.88 | 31.75 |
| | LOF | 106.56 | 30.31 | 0.00 |
| | Learned filter | 142.50 | 41.12 | 13.69 |
| 0.75 | $\text{CoreSet}_{\text{af}}$ | 138.75 | 38.69 | 5.69 |
| | $\text{CoreSet}_{\text{xf}}$ | 119.00 | 41.56 | 35.94 |
| | $\text{Random}_{\text{xf}}$ | 26.31 | 8.88 | 31.75 |
| | LOF | 99.38 | 25.44 | 0.00 |
| | Learned filter | 133.81 | 37.31 | 7.25 |
| 1.00 | $\text{CoreSet}_{\text{af}}$ | 136.62 | 38.12 | 6.25 |
| | $\text{CoreSet}_{\text{xf}}$ | 121.06 | 41.12 | 35.31 |
| | $\text{Random}_{\text{xf}}$ | 26.31 | 8.88 | 31.75 |
| | LOF | 101.19 | 25.94 | 0.00 |
| | Learned filter | 134.25 | 36.88 | 7.25 |

tions and normalize the resulting vectors. Labelable vectors, labels, row identities and failure counts stay fixed within each seed. The narrower failure groups remain locally distinct even when their centers coincide with rare-class centroids.

**Selection and endpoint.** All paired arms start with the same 30 items sampled from the full pool and take five rounds of 20 picks. Every inspection is charged, including initialization. Initial acquisitions do not count toward discovery AUC; a class encountered again in a later round does count. Selectors skip inspected items and continue after wholly failed rounds. We run anchored and skip-only CoreSet first, followed by random, LOF and the existing learned labelability filter, for 400 trajectories. The learned filter retains its established class-balanced logistic regression, threshold and fill behavior; until both feedback classes are observed it uses skip-only CoreSet. Selectors see embeddings and inspected outcomes, never uninspected class labels. The rare set stays fixed. Table D.1 reports all arms, including final discoveries and failed picks.

**Analysis and scope.** For each seed we subtract skip-only discovery AUC from anchored discovery AUC at each level. The primary exploratory contrast compares that advantage at zero and full interpolation using the exact paired one-sided sign-flip test. We report all levels with paired percentile bootstrap intervals from 10,000 resamples; these intervals are descriptive and unadjusted. The mean advantage remains positive at full interpolation, although individual pools can favor skipping. The ten-neighbor failure proportion is 1 and the rare-item proportion is 0 for every failure at all levels. This result cannot establish that shared local neighborhoods cause the attenuation. The supplementary evidence preserves the frozen design, all base pools, initialization and picks, input and source hashes, and a replay checker in `scripts/mechanism/run_boundary.py`.

**Descriptive camera-trap neighborhoods.** We use the original iWildCam and Serengeti manifest and embedding hashes recorded by the frozen runs. For each unlabelable item, we compute its ten nearest other items by cosine distance, resolving ties by row index. The fractions in Section 4.4 av-

Table E.1: **Exploratory detector filtering losses and computation.** Removed and Rare images are mean numbers of remaining candidates rejected after the shared initialization. Rare classes lost counts classes with at least one post-initial candidate before filtering but none afterward; the original evaluation denominator is unchanged. Inference is wall time for processing all 15,000 images, including decoding and output, excluding model loading. Unused is the maximum unused inspection budget across runs.

| Pool | Removed | Rare images | Rare classes lost | Inference (min) | Unused |
|---|---|---|---|---|---|
| iWildCam | 7707.3 | 12.9 | 6.0 | 27.1 | 0 |
| Serengeti | 10210.5 | 1.0 | 1.0 | 32.4 | 0 |

erage equally over all unlabelable items. Per-seed summaries for the failures selected by anchored CoreSet, skip-only CoreSet, random and LOF are also saved. These analyses reuse the original datasets and runs; they provide no independent validation and do not change the preregistered verdicts.

## E EXPLORATORY PRETRAINED-DETECTOR COMPARISON

**Frozen design and evaluation.** Before detector results were available, we fixed MegaDetector v5a, checkpoint `md_v5a.0.0.pt`, and an animal-confidence threshold of 0.1 (agentmorris, 2022). An image remains a candidate if it contains at least one animal detection at or above that threshold. We score every image once at an inference size of 1,280 pixels, but apply the filter only after the original initial pool has been inspected. The same initial rows are used in all six arms within seed, with 201 inspections on iWildCam and 88 on Serengeti, followed by ten batches of the same size. The total inspection budgets are therefore 2,211 and 968, including initialization. These inspections include failures; initial discoveries do not enter the endpoint, although a later acquisition of the same class counts. The original rare-class sets, 87 and 15 classes with at most five labelable images, never change after filtering. Rare classes made unreachable by the filter remain missed. Missing image IDs, failed inference and invalid confidence values are errors, not empty-frame predictions. An exhausted candidate pool leaves unused budget and carries discovery counts forward.

**Paired arms and reproducibility.** We compare anchored CoreSet, skip-only CoreSet and skipping random with and without filtering for seeds 0 to 15. Both CoreSet variants retain successful initial items as anchors even if the detector rejects them; only anchored CoreSet also uses failed inspected items as anchors. All inspected rows remain excluded from further selection, and the filter only removes uninspected candidates. Each arm skips inspected items and continues after wholly failed rounds. The frozen selector implementation and original input hashes are preserved in a standalone runner. We save detector scores, image hashes, every initialization and pick, per-round discoveries and failed picks. Filtered and unfiltered arms are rerun together. All unfiltered discovery trajectories match the original runs, although selected row identities differ in 36 CoreSet trajectories under numerical ties; these differences are recorded without changing geometry or tie behavior. The detector comparison uses the new paired runs and does not revise the preregistered H2 analysis. Paired intervals are descriptive percentile bootstrap intervals from 10,000 resamples with bootstrap seed 0, conditional on these two fixed pools and initialization seeds.

**Provenance and scope.** The checkpoint, inference dependencies and input files are hashed in the supplementary evidence. MegaDetector's documented training data include Snapshot Serengeti; exact overlap with the images used here was not established, and overlap with iWildCam is unknown. This is a comparison with an available pretrained filter, not a held-out detector-generalization test. Animal detection also differs from our annotation outcome: an unknown bird or bat may be detected yet remain unlabelable under the camera-trap protocol. Inference cost is additional to the charged human-inspection budget. The existing drone-pool logistic-regression comparison learns only from inspected feedback and uses simulated failures, so it tests a different intervention.

Both pools complete all planned inspections without candidate exhaustion. Inference uses an NVIDIA H100 80GB HBM3 MIG 1g.10gb partition. Model loading additionally takes 4.9 seconds on iWildCam and 4.7 seconds on Serengeti. The sum-

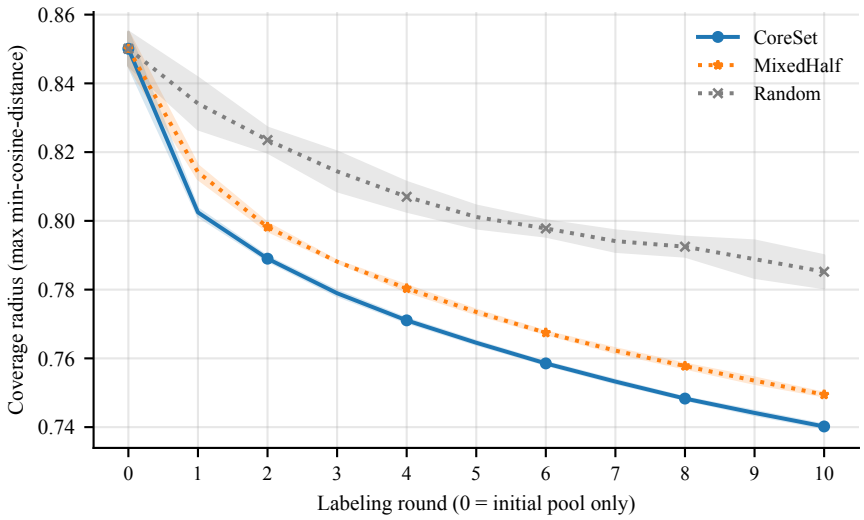



Figure F.1: **Coverage radius per round** on a synthetic long-tail dataset ($n = 5{,}000$, Zipf $s = 1.0$, $d = 128$). The radius is the largest cosine distance from a point to its nearest labeled point. Mean over $n_s = 8$ seeds, $\pm 1\sigma$ shaded. CoreSet (blue) shrinks the radius fastest by construction. Random (gray) is slowest.

mary includes final discoveries, post-initial failures and failures including initialization, every paired discovery-AUC difference, and unadjusted exploratory exact sign-flip tests. None of these tests enters a confirmatory family. The supplementary evidence is in `data/evidence/exploratory/detector`; `check_detector_evidence.py` recomputes outcomes from saved picks and `make_detector_table.py` regenerates the two tables.

# F Supporting analyses on synthetic data

The three analyses below probe the mechanism and the operating range of the main result on synthetic long-tail datasets. `scripts/make_figures.py` regenerates each one from the library's synthetic generator.

## F.1 Coverage radius

CoreSet's greedy $k$-center objective directly minimizes the coverage radius. The coverage radius is the largest cosine distance from any unlabeled point to its nearest labeled neighbor. Figure F.1 plots this radius on a synthetic long-tail dataset over $K = 10$ rounds and $n_s = 8$ seeds. The dataset has $n = 5{,}000$ points, 50 Zipf-distributed classes and $d = 128$. CoreSet shrinks the radius fastest, the hybrid comes second, and random is slowest. This fits a coverage explanation of the discovery advantage on real data, but it does not identify the cause.

## F.2 Budget allocation by rarity

We sort every pick from a 10-round simulation into three buckets by class size. The buckets are *rare* ($\leq 5$ samples), *uncommon* (6–20) and *common* ($> 20$). The simulation has $n = 5{,}000$, 200 classes, Zipf $s = 1.8$ and $k = 200$. Figure F.2 reports each selector's budget share per bucket. CoreSet spends a larger share on rare-class regions than MixedHalf or Random, and a smaller share on the dense common core. In this synthetic construction, rare classes occupy sparse regions. In other embeddings, sparse regions can hold outliers or within-class variation instead.

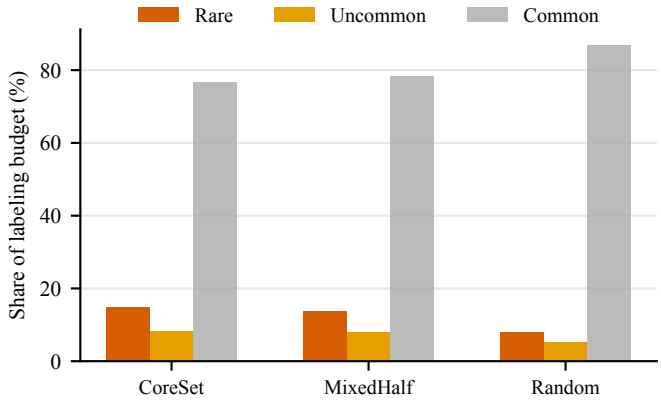



Figure F.2: **Budget allocation by class-frequency bucket** on a synthetic long-tail dataset ($n = 5{,}000$, 200 classes, Zipf $s = 1.8$). CoreSet directs a larger share of its picks to rare-class regions than MixedHalf or Random.

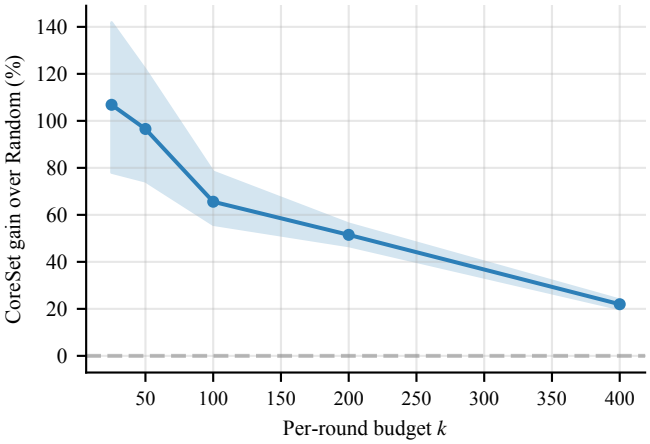



Figure F.3: **CoreSet gain over Random as a function of per-round budget** $k$, on a synthetic long-tail dataset ($n = 5{,}000$, 200 classes, Zipf $s = 1.5$). Shaded: bootstrap 95% CI.

### F.3 Budget sensitivity

Figure F.3 sweeps $k \in \{25, 50, 100, 200, 400\}$ on a synthetic long-tail dataset. The dataset has $n = 5{,}000$, 200 classes and Zipf $s = 1.5$, with $n_s = 8$ seeds and 10 rounds. CoreSet beats Random at every budget, and the 95% bootstrap CI excludes zero at each one. In this synthetic setting, the gain is largest at small $k$ and shrinks at large $k$ as the pool saturates.

## G A geometric model of discovery

We prove idealized versions of the three regularities of Appendices C.2 to C.4 in a stylized model. `scripts/theory_check.py --check` simulates each statement and verifies its assumptions. It uses the library's CoreSet, TypiClust and ProbCover where they apply, and it exits nonzero if a bound fails.

**Model.** $V$ is the finite set of labelable rows with metric $\rho(x, y) = \|x - y\| = \sqrt{2(1 - \langle x, y \rangle)}$ on unit vectors. The metric increases with cosine distance. So farthest-first and nearest-neighbor rank-

ings agree with the cosine rankings, a cosine radius $\delta$ is a $\rho$-radius $\sqrt{2\delta}$, and TypiClust's squared distance is $\rho^2$. Classes partition $V$ into $V_c$ with $n_c = |V_c|$, and $d(x, A) = \min_{a\in A} \rho(x, a)$. A set $S$ *discovers* $c$ if $S \cap V_c \neq \emptyset$, and $U(S)$ is the set of undiscovered classes. *Farthest-first* (FF) starts from $S_0 \neq \emptyset$ and picks $x_t \in \arg\max_{x\notin S_{t-1}} d(x, S_{t-1})$ (Gonzalez, 1985). Under REMOVED, CoreSet's picks over $R$ rounds of $k$ are the first $Rk$ terms of one FF sequence. Let $D_{\text{in}} = \max_c \operatorname{diam} V_c$ and $D_{\text{out}} = \min_{c\neq c'} d(V_c, V_{c'})$. **A1** (separation): $D_{\text{in}} < D_{\text{out}}$. Real embeddings violate A1 (Appendix P), so we also use two quantities that need no assumption. The *isolation* $\iota_c = \max_{x\in V_c} d(x, V \setminus V_c)$ is the nearest-neighbor distance for a single-row class. The *packing number* $M(\Delta)$ is the size of the largest $\Delta$-separated subset of $V$. A1 implies $\iota_c \geq D_{\text{out}}$.

**Proposition 1** (Distance-seeking)**.** *(a) Under A1 each FF pick lies in an undiscovered class while one remains, so after $m$ picks FF has discovered $\min(m, |U(S_0)|)$ classes, whatever the $n_c$. (b) Without A1, each $c \in U(S_0)$ is discovered within $\tau_c \leq M(\iota_c) - 1$ picks. (c) Let $s_\kappa(x)$ be the distance from a pool row $x$ to its $\kappa$-th nearest other pool row. Under A1, every pool row of a class with at most $\kappa$ pool rows has larger $s_\kappa$ than every pool row of a larger class. So the top $m$ rows by $s_\kappa$ span at least $\lceil m/\kappa \rceil$ classes, for $m$ at most the number of rows in such small classes.*

*Proof.* (a) A row of $c \in U(S_{t-1})$ has $d(\cdot, S_{t-1}) \geq D_{\text{out}}$; a candidate of a discovered class is within $D_{\text{in}} < D_{\text{out}}$ of a labeled row of its class. (b) Let $x^\star \in V_c$ attain $\iota_c$. Before $c$ is discovered $S_{t-1} \cap V_c = \emptyset$, so $d(x_t, S_{t-1}) \geq d(x^\star, S_{t-1}) \geq \iota_c$. Hence $\{s_0, x_1, \ldots, x_{\tau_c-1}, x^\star\}$, with $s_0 \in S_0$, is $\iota_c$-separated with $\tau_c + 1$ points. (c) A small-class row has at most $\kappa - 1$ same-class neighbors, so $s_\kappa \geq D_{\text{out}}$; a large-class row has $\kappa$ within $D_{\text{in}}$. Each small class contributes at most $\kappa$ rows. □

**Lemma 1** (Random, Hurlbert, 1971)**.** *$m$ uniform draws without replacement from $N'$ pool rows discover $\sum_{c\in U(S_0)}[1 - \binom{N'-n_c}{m}/\binom{N'}{m}] \leq \sum_{c\in U(S_0)} \min(1, mn_c/N')$ classes in expectation. The equality is the hypergeometric miss probability, and the bound is a union bound over the $n_c$ rows.*

So under A1 and $m \leq |U(S_0)|$, random finds at most a fraction $w = \sum_{c\in U(S_0)} n_c/N'$ of FF's count. Restricting the sum to rare classes bounds random's expected rare discoveries per pick. On BCI the 157 rare classes hold 333 rows. With $N' = 3{,}080$ pool rows after a 200-row initial set, the bound is below 0.11. Under A1, every FF pick discovers a new class.

*Remark* 1 (Overlap). By Proposition 1(b), after $m$ picks FF has discovered every class with $M(\iota_c) \leq m + 1$, whatever its mass. A class nested inside another has small $\iota_c$ and no guarantee, in line with Kurinchi-Vendhan & Beery (2026). In $\mathbb{R}^q$, rows in a ball of radius $R$ have $M(\Delta) \leq (1 + 2R/\Delta)^q$, by counting disjoint $\Delta/2$-balls. So the guarantee weakens with dimension. This fits whitening, which raises intrinsic dimension from 9 to 35 and loses 16 points (Appendix P). The bound is loose: on overlapping planar data the planar packing bound $4(1 + \iota_c)^2/(\pi\iota_c^2)$ certifies at most 2 classes in 100 picks, while FF finds 18 to 28.

**Proposition 2** (Density-seeking)**.** *Take one TypiClust round (Appendix B) with partition $\Pi$. Clusters of at most 5 rows are dropped. A visit to cluster $Q$ with $q$ candidate rows picks the row that minimizes $\bar{r}^2$, the mean squared distance to its $K = \min(20, \lfloor q/2 \rfloor)$ nearest candidate rows in $Q$. Assume no fallback draw occurs. (a) A class all of whose candidate rows lie in clusters of at most 5 rows is not picked; if $\Pi$ is class-pure, no class with $n_c \leq 5$ is picked. (b) Under A1, if $Q$ has at least $K + 1$ candidate rows of class $a$ and $j \geq 1$ of class $c$, and $(K - j + 1)D_{\text{out}}^2 > KD_{\text{in}}^2$ ($D_{\text{out}} > 1.12\, D_{\text{in}}$ suffices for $K = 20$, $j \leq 5$), the pick is not in $c$. (c) ProbCover with $\rho$-radius $\delta < d(V_c, V \setminus V_c)$ picks a row of $c$ only when the largest number of uncovered rows in any $\delta$-ball is at most $n_c$.*

*Proof.* (a) Picks come only from clusters of more than 5 rows. (b) A row of $c$ has at most $j - 1$ of its $K$ neighbors in $c$ and the rest at least $D_{\text{out}}$ away, so $\bar{r}^2 \geq (K - j + 1)D_{\text{out}}^2/K > D_{\text{in}}^2$, while a row of $a$ has $K$ neighbors in $a$ and $\bar{r}^2 \leq D_{\text{in}}^2$. (c) A $\delta$-ball around a row of $c$ lies in $V_c$, and ProbCover picks a ball of maximum gain. □

*Remark* 2. Proposition 2 shows that the density terms defer small classes. The ablations of Appendix C.4 remove exactly the mechanisms in (a) and (b). The proposition does not show that these selectors do no better than random, which remains an empirical question. The model favors ProbCover: with $D_{\text{in}} \leq \delta < D_{\text{out}}$ it finds one class per pick, in decreasing size. This fits ProbCover's gain at a larger label-free radius. We give no bound for the $k$-means++ variant.

**Unlabelable rows.** An *unlabelable item* is an image that returns no class label when picked, and a pick of one is a *failed pick*. Add unlabelable rows $O = O_1 \cup \cdots \cup O_J$ under NAIVE. Within a round, CoreSet's reference set $T \supseteq S$ grows with each pick. At the round's end, labelable picks join $S$, while unlabelable picks are discarded and stay candidates. Let $r_0 = \max_{x \in V} d(x, S_0)$ and $D_O = \max_j \operatorname{diam} O_j$. **A2** (far, tight groups): $d(O_j, (V \cup O) \setminus O_j) > \max(r_0, D_O)$ for all $j$.

**Proposition 3** (Unlabelable rows)**.** *(a) Under A2 the first* $\min(k, J)$ *picks of every round lie in distinct groups. If* $J \geq k$*, FF discovers nothing in any round. Random draws* $k|V \setminus S|/(|V \setminus S| + |O|)$ *labelable rows per round in expectation. (b) Let the selector remember its failed picks* $F$ *and skip them: they are never candidates again (*SELECTABLE*).* Skip-only *FF keeps* $S$ *as the round's initial reference set.* Anchoring *FF starts from* $S \cup F$*, so a row near a failed pick counts as covered. Assume A1, A2,* $D_O < D_{\text{out}}$ *and* $J \leq k$*. In* $R$ *rounds, skip-only FF discovers exactly* $\min(Rk - W, |U(S_0)|)$ *classes with* $W = \sum_j \min(|O_j|, R)$*. Anchoring FF discovers exactly* $\min(Rk - J, |U(S_0)|)$*. Anchoring thus saves* $W - J = \sum_j (\min(|O_j|, R) - 1)$ *picks. It saves none when every group is a single row, and* $(R-1)J$ *when every group has at least* $R$ *rows.*

*Proof.* (a) A group disjoint from $T$ has all rows farther than $\max(r_0, D_O)$ from $T$. A labelable row is within $r_0$ of $S_0 \subseteq T$, and a row of a visited group is within $D_O$ of $T$. Random draws uniformly from $(V \setminus S) \cup O$. (b) Call a class open while $T$ holds none of its rows. If a class is open, then $r_0 \geq D_{\text{out}}$, so by A1 and A2 its rows are at least $D_{\text{out}}$ from $T$. Rows of other classes are within $D_{\text{in}}$ of $T$, and rows of a group that meets $T$ are within $D_O$. Hence, while a class is open, each round first picks one row from every group not yet within $D_O$ of $T$, as in (a). It then discovers a new class with every pick. For skip-only FF the round starts from $S$, and each earlier round removed one row of every group from the candidates. So round $r$ spends $|\{j : |O_j| \geq r\}| \leq k$ picks on $O$, and $W$ picks in all. For anchoring FF, round 1 spends $J$ picks and puts a row of every group in $F$, so later rounds spend none. If every class is discovered before round $R$ ends, the count is $|U(S_0)|$. Otherwise all $Rk$ picks are made with a class open, and the count is $Rk$ minus the picks spent on $O$. □

*Remark* 3. A2 is a stylization. BCI's 4,437 unlabelable items are images never labeled to species, not images known to lack a class. A probe separates labeled images from unlabelable items with ROC AUC near 0.71 (Appendix H). The model reproduces the mechanism of the runs scored flat in Appendix I: FF prefers the rows far from the labeled data, and discarded picks recur every round. The simulator stops a NAIVE run once a whole round returns no label, and it counts no further discovery (Table M.2). Part (b) predicts that anchoring failed picks helps in proportion to how tightly the unlabelable rows cluster. It helps little when they are spread out, since a failed pick then covers few other unlabelable rows. It saves $(R-1)$ picks per group when they form tight groups of at least $R$ rows. For $J > k$, the same argument shows that anchoring stalls for about $J/k$ rounds, until every group has a failed pick. Skip-only FF stalls as long as $k$ groups have rows left. On BCI ($R = 10$, $k = 200$), $\text{CoreSet}_{\text{xf}}$ spends 1,354 to 1,423 of its 2,000 picks on unlabelable items, and $\text{CoreSet}_{\text{af}}$ spends 1,294 to 1,352 (`data/evidence/labelability`). Anchoring thus saves 50 to 71 picks and adds 0.1 to 6.4% in discovery AUC (Appendix I). This fits loosely grouped unlabelable items, not a few tight groups. We read this after the runs, not as a prediction made before them, and the runs do not test A2. H2a is the preregistered test of part (b)'s comparison on D1 and D2 (Table 1).

## H REFERENCE DATASET RATIONALE

The rarity definition uses observed expert-label counts without artificial class balancing. These sample counts do not establish rarity in the wider forest population. The labeled sample is also not a random draw from the images. A cross-validated logistic regression on the frozen embedding separates labeled images from unlabelable items. Its ROC AUC is 0.71 (DINOv3), 0.72 (DINOv2) and 0.72 (PlantCLEF), where 0.5 would mean indistinguishable. Labeling is also concentrated in time: 28 of the 62 acquisition dates contribute no labeled image, and 10 dates hold half of them (`scripts/labelled_sample_probe.py`). This separability is why the covered anchors move geometry-based selectors in a direction we cannot predict (Appendix C). The 157 rare classes are rare in a convenience sample. A campaign that labeled different flights could define a different rare set. Repeated views of one tree are not grouped. The endpoint counts distinct classes, so a duplicated crown cannot be counted twice. Each image is one crop from its own drone frame. No two rows are near-identical in any encoder. The largest cosine

similarity between different rows is 0.967 (DINOv3), 0.984 (DINOv2) and 0.978 (PlantCLEF) (`scripts/near_duplicate_probe.py`). Of the 157 rare classes, 76 have a single labeled image. Of the other 81, 10 have all their images from one flight date. The image keys carry no tree identifier. So some rare-class images may show the same tree from overlapping frames or on different dates. That would make such a class easier to reach for every selector.

# I COVERED RESULTS

COVERED passes the unlabelable items to the selector as labeled anchors, and Appendix I.1 shows the code. Tables I.1 and I.2 give the audit under COVERED on the core family. The $p$-values in these two tables are SciPy's paired Wilcoxon, as in the original COVERED analysis (Appendix X). The exact sign-flip $p$ for CoreSet against NovelMargin on DINOv3 is 0.118 (Table C.1). `check_claims.py --check` rebuilds every row of both tables from the saved COVERED evidence and fails on any difference.

Table I.1: **COVERED, DINOv3 ViT-L/16**, $K = 10$, $k = 200$, $n_s = 16$. "vs. Random" is each selector's own gain over random (Appendix W). "CoreSet gain" is CoreSet against that row, with SciPy's one-sided Wilcoxon $p$ and the Holm decision at $\alpha = 0.025$, $m = 6$. Brackets are 95% bootstrap intervals. $^*$: random is tested alone for the superiority claim, outside the Holm family. The joint claim fails because CoreSet does not beat NovelMargin.

| Selector | AUC | vs. Random | CoreSet gain | Wilcoxon $p$ | Holm |
|---|---|---|---|---|---|
| CoreSet | 1066 | +39.11% [+37.00, +41.32] | — | — | — |
| NovelMargin | 1061 | +38.41% [+36.21, +40.66] | +0.51% [−0.28, +1.24] | 0.06 | × |
| MaxHerding | 1050 | +37.04% [+34.79, +39.45] | +1.51% [+1.10, +1.94] | $3.3\times10^{-4}$ | ✓ |
| MixedHalf | 962 | +25.51% [+22.83, +28.14] | +10.84% [+9.69, +12.21] | $2.2\times10^{-4}$ | ✓ |
| TypiClust | 803 | +4.85% [+2.04, +7.60] | +32.67% [+29.53, +36.07] | $2.2\times10^{-4}$ | ✓ |
| ProbCover | 799 | +4.29% [+1.83, +6.73] | +33.39% [+30.97, +35.69] | $2.2\times10^{-4}$ | ✓ |
| Random | 766 | — | +39.11% [+37.00, +41.32] | $2.2\times10^{-4}$ | ✓$^*$ |
| $k$-means++ | 751 | −1.95% [−4.22, +0.31] | +41.88% [+39.25, +44.60] | $2.2\times10^{-4}$ | ✓ |

Table I.2: **COVERED across encoders.** CoreSet AUC and its gain over NovelMargin, MaxHerding, MixedHalf and random ($n_s = 16$). The bracketed interval is the 95% bootstrap CI on the comparison that decides the joint claim. Every other comparison is rejected under Holm with SciPy Wilcoxon $p \le 3.3\times10^{-4}$.

| Embedding | Dim | AUC | vs. NovelMargin | vs. MaxHerding | vs. MixedHalf | vs. Random | Joint |
|---|---|---|---|---|---|---|---|
| DINOv3 ViT-L/16 (224) | 1024 | 1066 | +0.51% [−0.28, +1.24] | +1.51% | +10.84% | +39.11% | FAIL |
| DINOv2-Giant (224) | 1536 | 1077 | −0.83% [−1.51, −0.15] | +3.17% | +13.51% | +40.54% | FAIL |
| PlantCLEF2024 (672) | 768 | 1116 | −0.53% [−1.03, +0.01] | +2.32% | +12.48% | +45.61% | FAIL |

**Per-round trajectories.** Figure I.1 shows the COVERED trajectories. CoreSet leads MixedHalf from round 1 on every encoder. NovelMargin leads CoreSet early. In mean cumulative rare classes, CoreSet trails NovelMargin in rounds 1 to 3 on DINOv3 and leads from round 4. On DINOv2 it trails in rounds 1 to 4, leads in rounds 5 to 9, and trails by 0.44 rare classes at round 10. On PlantCLEF it trails in rounds 1 to 5 and leads from round 6. NovelMargin's edge under this pool convention is therefore front-loaded. Under REMOVED, CoreSet leads in every round on DINOv3 and PlantCLEF. On DINOv2 it ties at round 2 and trails by 0.62 at round 3 (Figure J.1).

**Round 1.** At $K = 1$ under COVERED, CoreSet finds $38.2 \pm 1.5$ rare classes on DINOv3, against $32.8 \pm 3.2$ for MixedHalf. Values are mean $\pm$ sample standard deviation over 16 seeds. On all three encoders, the round-1 gap exceeds 1.7 times the larger of the two sample standard deviations. On this pool, the hybrid's random half bought no exploration benefit, even in the cold-start round.

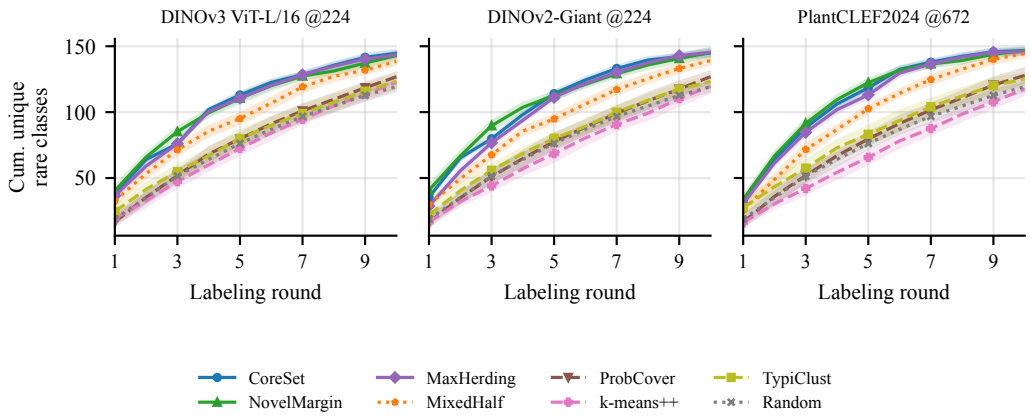



Figure I.1: **Per-round trajectories on three encoders, COVERED.** Cumulative unique rare classes, mean over $n_s = 16$ seeds, $\pm 1\sigma$ shaded. CoreSet leads the hybrid from round 1 on every panel. NovelMargin leads CoreSet through round 3 (DINOv3), 4 (DINOv2) or 5 (PlantCLEF).

**Annotation efficiency.** For each seed, we find the first round at which the challenger's cumulative rare-class count reaches the baseline's final-round count. Under COVERED on DINOv3, CoreSet reaches random's 10-round discovery level by round 6.1 on average over 16 seeds. This saves 39% of the acquisition-round budget, not counting the shared initial pool. The hybrid needs 7.6 rounds.

**MixedRatio.** A parametrized hybrid picks $\mathrm{round}(k\rho)$ items with CoreSet and the rest at random. Under COVERED on DINOv3, CoreSet beats the hybrid by $+21.28\%$, $+10.84\%$ and $+4.15\%$ at $\rho = 0.25$, 0.5 and 0.75 (each $p \leq 2.2 \times 10^{-4}$). Every hybrid beats random.

**Novel-only endpoint.** The discovery AUC counts a rare class once a round acquires it, even when the initial pool already held an example. The initial pools hold 19.5 rare classes on average. Counting only classes absent from the initial pool leaves every pick unchanged and lowers random's AUC from 766 to 679 (`run_novel_only_n16.py`). Under COVERED, CoreSet's gain over random becomes $+41.70\%$ $[+39.63, +43.94]$ on DINOv3, $+43.33\%$ on DINOv2 and $+48.40\%$ on PlantCLEF. On DINOv3 its lead over NovelMargin grows from $+0.51\%$ to $+1.18\%$ $[+0.51, +1.84]$, and the joint claim passes. On DINOv2 and PlantCLEF, NovelMargin stays nominally ahead ($-0.45\%$ and $-0.21\%$).

**NAIVE.** Under NAIVE on DINOv3, random spends 733 of its 2,000 picks on labeled rows, and CoreSet spends 104. The simulator stops a run once a whole round lands on unlabelable items, since the labeled set no longer changes. We score such a run as flat from that round on. This applies to most CoreSet, MaxHerding, NovelMargin and TypiClust runs. In the SELECTABLE runs of Appendix C.2, also treating failed picks as covered anchors adds 0.1 to 6.4% in mean discovery AUC. Proposition 3(b) ties this gain to how tightly the unlabelable items cluster. In the separated-group model, a failed anchor avoids spending more budget on the same unlabelable group; this is a sufficient account under the model assumptions, not a necessary condition in arbitrary embeddings. We do not test this difference on BCI. There the result rests on SELECTABLE CoreSet against SELECTABLE random, not on anchoring. H2a tests anchoring, preregistered, on D1 and D2 (Table 1). SELECTABLE CoreSet's gains over random coincide on DINOv3 and DINOv2 because neither random arm depends on the encoder. Its mean discovery AUC is 595.25 on both encoders by coincidence. Its per-seed values differ on 15 of 16 seeds (seed 0: 556 and 572). COVERED marks every unlabelable item as an anchor from the start. Under COVERED, CoreSet gains $+0.51\%$ over NovelMargin on DINOv3 ($p = 0.118$, TOST-equivalent). NovelMargin is nominally ahead on DINOv2 and PlantCLEF ($-0.83\%$ and $-0.53\%$).

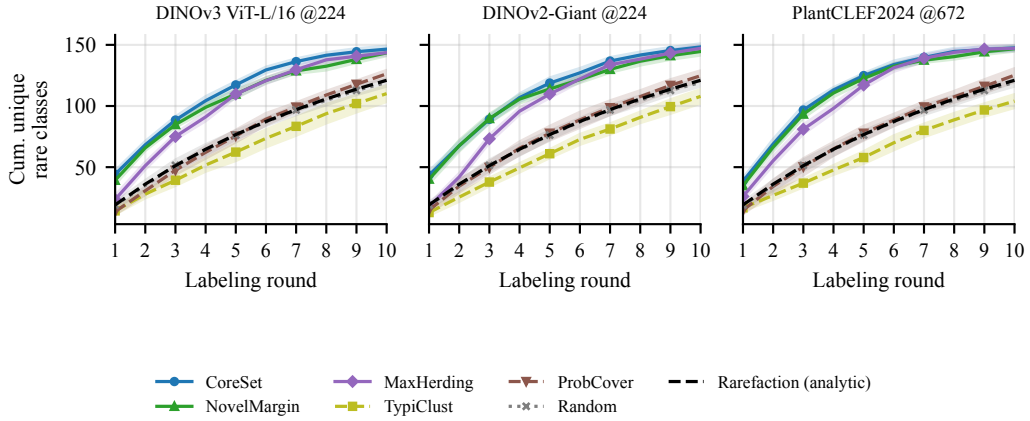



Figure J.1: **Cumulative distinct rare classes by round, REMOVED**, one panel per encoder. Mean over 16 seeds, ±1 standard deviation shaded. The dashed line is random's analytic rarefaction expectation. Distance-seeking CoreSet and NovelMargin rise fastest. Density-seeking ProbCover tracks random, and TypiClust falls below it.

### I.1 ILLUSTRATIVE CODE

The harness ships a regression test for COVERED. The test captures the selector's input each round, and it fails if a held-out index is missing from the labeled or covered mask.

```
# src/labelfirst/eval/simulate.py
# Rows without ground truth remain visible as covered anchors.
# Mark them as labeled to exclude acquisition while retaining their
# effect on the coverage geometry; their labels remain unavailable.
held_out = np.where(~labeled_mask)[0]
for r in range(rounds):
    labeled_idx = np.asarray(
        sorted(labeled_set | set(held_out.tolist())), dtype=np.int64
    )
    picks_res = strat.select(X, labeled=labeled_idx,
                             k=min(k_per_round, len(pool_set)))
    picks = [int(p) for p in picks_res.picks if int(p) in pool_set]
```

## J SENSITIVITY ANALYSES UNDER REMOVED

Both analyses rerun the core family on DINOv3 under REMOVED with $n_s = 16$. The scripts are `run_pool_block_loo_sensitivity.py` and `run_rare_threshold_sensitivity.py`, and `sensitivity_check.py` checks them. Their tests are the library's paired Wilcoxon with Holm, not the exact sign-flip test. *Pool leave-one-block-out:* we split the acquisition dates into five consecutive blocks and drop each block's labeled rows in turn. The joint claim passes in every block. CoreSet's gain over random is +31.74%, +31.28%, +35.08%, +38.53% and +29.17%. Its gain over NovelMargin is +4.27%, +3.63%, +4.05%, +3.69% and +3.28%. *Rare threshold:* the joint claim passes at thresholds of 10 and 2. At 10 (185 rare classes), CoreSet gains +37.04% over random and +4.49% over NovelMargin. At 2 (106 rare classes), it gains +66.23% and +9.04%. *TypiClust:* the rare threshold does not change its sign. It falls 19.3% below random at a threshold of 2 and 11.0% at 10, against 14.1% at the main threshold of 5, and trails random on all 16 seeds at both. Its sign does depend on the pool convention: under COVERED it gains 4.9 to 9.6% over random, ahead on 13 to 14 of 16 seeds, while under REMOVED it falls 14 to 18% below (Table Q.1). The seed counts are in `data/evidence/*_n16_runs.json`.

## K H1 CONTRAST APPLIED POST HOC TO THE DRONE POOL

Our preregistration defines, for each seed $s$, $d_s = \overline{\log(A_s+1)}_{F_{\text{dist}}} - \overline{\log(A_s+1)}_{F_{\text{dens}}}$. Here $A_s$ is the discovery AUC under REMOVED. The distance family $F_{\text{dist}}$ is CoreSet, $k$NN distance, LOF, Isolation Forest, NovelMargin and MixedHalf. The density family $F_{\text{dens}}$ is TypiClust, ProbCover and $k$-means++. The contrast reports $\exp(\bar{d}) - 1$ with an exact sign-flip test on the $d_s$. `scripts/h1_bci_contrast.py` applies it to the saved drone-pool runs. These runs share seeds and initial pools across the three evidence files the script reads. The script writes the per-seed $d_s$ and their signs to `data/evidence/h1_bci`. The drone pool is exploratory in the preregistration, so this contrast enters no confirmatory decision. Two variants test whether one arm drives the result. One drops $k$-means++, the surrogate we added. The other replaces purity-radius ProbCover with the label-free radius of Appendix C.4.

Table K.1: **H1 contrast applied post hoc to BCI**, REMOVED, $n_s = 16$: $100(\exp(\bar{d})-1)$, in percent. Every variant is positive on all 16 seeds of every encoder, exact sign-flip $p = 2^{-16}$.

| Density family | DINOv3 | DINOv2 | PlantCLEF |
|---|---|---|---|
| Preregistered (TypiClust, ProbCover, $k$-means++) | +38.2 | +43.5 | +49.5 |
| Without $k$-means++ | +40.7 | +44.4 | +49.6 |
| Label-free ProbCover radius | +30.5 | +36.1 | +39.7 |

## L DRONE-POOL TESTS AND THE FULL REMOVED COMPARISON

**Equivalence, power and the extended family.** A failed joint test shows neither equivalence nor the absence of an effect. For every comparison we therefore also run TOST (Schuirmann, 1987), two one-sided exact sign-flip tests against a margin of $\pm 2\%$ of the random arm's mean endpoint, and report the minimum detectable effect and TOST power (Table M.4). BCa intervals (Efron, 1987) are in Table M.2. An alternative beating CoreSet is an exploratory direction, reported only as a $p$-value. The *extended* family of ten replaces $k$-means++ with gradient BADGE and adds $k$NN distance, LOF, Isolation Forest and NNDB. The mechanism ablations (Appendix C.4) and sensitivity analyses use the library's paired Wilcoxon test instead, except CoreSet against label-free ProbCover.

**The full comparison.** Table L.1 gives every selector under REMOVED. $k$NN distance is the strongest alternative on every encoder. It is 1.1 to 1.5% behind CoreSet on DINOv3 and PlantCLEF and tied with it on DINOv2 ($+0.64\%$ $[-0.11, +1.35]$, $p = 0.060$, not TOST-equivalent). The joint test against the extended family thus passes on two encoders and fails on DINOv2 because of this one comparison. The ProbCover radius in Table L.1 comes from a purity rule that uses the ground-truth class count, so the main ProbCover comparison uses a label-free radius (Appendix C.4). CoreSet leads it by $+22.64\%$ on DINOv3, on all 16 seeds, with exact $p = 2^{-16}$ on every encoder; this comparison lies outside both families. CoreSet leads NovelMargin in every round on DINOv3 and PlantCLEF (Figure J.1).

**Robustness.** Under REMOVED the core-family joint test fails in two secondary cases (Table M.1): the final count $u_K$ on PlantCLEF and the early AUC on DINOv2, where NovelMargin (and on PlantCLEF $u_K$ also MaxHerding) is not separated from CoreSet. Date-block and rare-threshold analyses keep the joint verdict (Appendix J).

## M FULL SPECIFICATION TABLE AND POWER

Table M.2 gives every pool convention, endpoint and encoder with the core family (`audit_stats.py`). It adds BCa intervals and TOST for CoreSet against NovelMargin. Table M.4 gives the minimum detectable effect (MDE) at 80% power under REMOVED. It also gives the TOST power at a true difference of zero. Most comparisons have an MDE of 1 to 3%, so the study can resolve the gaps it reports. TOST power is low where the paired differences are noisy. So we do not read a failed superiority test as equivalence.

Table L.1: **REMOVED, discovery AUC** (3,280 labeled rows, $K = 10$, $k = 200$, $n_s = 16$). "vs. random" is each selector's own gain over random (Appendix W). "CoreSet gain" is CoreSet's gain over that selector. †: not rejected under Holm in its family (extended family; $k$-means++ is only in the core family). ProbCover uses its label-derived purity radius, a secondary comparison; the label-free radius is compared in the text. The last rows give the intersection-union verdict of the joint claim and the Holm rejection count; numbers in percent.

| | | DINOv3 | | DINOv2 | | PlantCLEF | |
|---|---|---|---|---|---|---|---|
| | Selector | vs. random | CoreSet gain | vs. random | CoreSet gain | vs. random | CoreSet gain |
| | CoreSet | +46.3 | — | +46.8 | — | +50.6 | — |
| Distance | $k$NN distance | +44.6 | +1.1 | +45.9 | +0.6† | +48.4 | +1.5 |
| Distance | NovelMargin | +38.7 | +5.5 | +42.4 | +3.1 | +47.4 | +2.2 |
| Distance | MixedHalf | +30.5 | +12.1 | +31.0 | +12.1 | +32.6 | +13.6 |
| Distance | LOF | +18.5 | +23.4 | +20.5 | +21.9 | +33.4 | +12.9 |
| Distance | Isolation Forest | +9.6 | +33.4 | +14.3 | +28.5 | +9.2 | +37.9 |
| Mixed | BADGE (gradient) | +37.4 | +6.4 | +38.2 | +6.2 | +41.3 | +6.5 |
| Mixed | MaxHerding | +33.5 | +9.5 | +33.5 | +10.0 | +41.7 | +6.3 |
| Mixed | NNDB | +6.4 | +37.5 | +2.9 | +42.6 | +4.1 | +44.7 |
| Density | ProbCover | +0.6 | +45.4 | +1.7 | +44.4 | +1.6 | +48.2 |
| Density | $k$-means++ | −1.9 | +49.2 | −6.2 | +56.6 | −8.9 | +65.3 |
| Density | TypiClust | −14.1 | +70.2 | −16.7 | +76.2 | −18.4 | +84.5 |
| | Joint, core (6) | PASS (6/6) | | PASS (6/6) | | PASS (6/6) | |
| | Joint, extended (10) | PASS (10/10) | | FAIL (9/10) | | PASS (10/10) | |

Table M.1: **Specification table.** CoreSet's gain over random in percent, and the Holm rejection count in the core (6) and extended (10) families. †: exact $p > \alpha$. Bold marks a passing intersection-union test of the joint claim. A dash marks baselines not run under that pool convention. Table M.2 adds the trapezoid and oracle-normalized AUCs, BCa intervals and TOST.

| | | DINOv3 | | | DINOv2 | | | PlantCLEF | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Convention | Endpoint | vs. random | Holm 6 | Holm 10 | vs. random | Holm 6 | Holm 10 | vs. random | Holm 6 | Holm 10 |
| NAIVE | Discovery AUC | −27.2† | 3/6 | — | −32.5† | 2/6 | — | −49.5† | 1/6 | — |
| | Final $u_K$ | −55.5† | 1/6 | — | −59.1† | 1/6 | — | −69.1† | 1/6 | — |
| | Early AUC (rounds 1–3) | +62.5 | 4/6 | — | +52.2 | 4/6 | — | +12.4† | 3/6 | — |
| COVERED | Discovery AUC | +39.1 | 5/6 | 5/10 | +40.5 | 5/6 | 5/10 | +45.6 | 5/6 | 5/10 |
| | Final $u_K$ | +21.3 | **6/6** | 4/10 | +21.5 | 4/6 | 3/10 | +23.1 | 5/6 | 4/10 |
| | Early AUC (rounds 1–3) | +68.9 | 5/6 | 5/10 | +71.1 | 5/6 | 5/10 | +76.1 | 5/6 | 5/10 |
| REMOVED | Discovery AUC | +46.3 | **6/6** | **10/10** | +46.8 | **6/6** | 9/10 | +50.6 | **6/6** | **10/10** |
| | Final $u_K$ | +22.9 | **6/6** | 8/10 | +24.4 | **6/6** | 7/10 | +23.7 | 4/6 | 6/10 |
| | Early AUC (rounds 1–3) | +90.4 | **6/6** | **10/10** | +89.5 | 5/6 | 8/10 | +93.4 | **6/6** | **10/10** |

### M.1 EXACT TEST $p$-FLOOR

A one-sided exact paired test on $n$ seeds has a discrete $p$-value distribution. With $n$ nonzero differences all favoring the same arm, the smallest attainable one-sided $p$ is $1/2^n$. Table M.5 lists this floor. It also lists how many simultaneous Holm-corrected comparisons can fire at one-sided $\alpha = 0.025$. At five seeds, no corrected claim can fire. The table illustrates resolution and does not guarantee power. Table M.4 gives power.

**Replaying five-seed practice on the saved runs.** Table M.6 recomputes each CoreSet comparison under COVERED on every five-seed subset of the sixteen paired seeds. It uses three decision rules. The first is the higher mean AUC. The second is an uncorrected one-sided paired Wilcoxon test at $p < 0.05$ in either direction. The third is Holm step-down. No simulation is rerun. For the five large-effect alternatives, the informal rules agree with the sixteen-seed result in every subset but one case. That case is MaxHerding on DINOv3, where the uncorrected test reaches significance in 68.8% of subsets. For the close comparison with NovelMargin, the outcome depends on which seeds were

Table M.2: **Full specification table**, core family, exact sign-flip tests. The TOST margin is $\pm 2\%$ of random's mean. $^{\dagger}$: CoreSet vs. random exact $p > \alpha$. Continued in Table M.3.

| | | vs. random | Joint, core | | CoreSet vs. NovelMargin | |
|---|---|---|---|---|---|---|
| Endpoint | Encoder | gain (%) | IUT | Holm | gain (%) [BCa 95% CI] | TOST |
| NAIVE | | | | | | |
| Discovery AUC | DINOv3 | $-27.2^{\dagger}$ | FAIL | 3/6 | $-8.8$ [$-14.5$, $-3.3$] | no |
| | DINOv2 | $-32.5^{\dagger}$ | FAIL | 2/6 | $-11.3$ [$-15.0$, $-7.4$] | no |
| | PlantCLEF | $-49.5^{\dagger}$ | FAIL | 1/6 | $-15.6$ [$-20.7$, $-11.6$] | no |
| Trapezoid AUC | DINOv3 | $-96.1^{\dagger}$ | FAIL | 0/6 | $-52.9$ [$-60.4$, $-43.8$] | no |
| | DINOv2 | $-96.4^{\dagger}$ | FAIL | 0/6 | $-53.6$ [$-60.3$, $-46.7$] | no |
| | PlantCLEF | $-97.6^{\dagger}$ | FAIL | 0/6 | $-45.1$ [$-52.2$, $-37.3$] | no |
| Final $u_K$ | DINOv3 | $-55.5^{\dagger}$ | FAIL | 1/6 | $-13.2$ [$-19.0$, $-7.6$] | no |
| | DINOv2 | $-59.1^{\dagger}$ | FAIL | 1/6 | $-16.0$ [$-20.5$, $-11.5$] | no |
| | PlantCLEF | $-69.1^{\dagger}$ | FAIL | 1/6 | $-18.8$ [$-24.1$, $-14.8$] | no |
| Early AUC (1–3) | DINOv3 | +62.5 | FAIL | 4/6 | +0.8 [$-4.9$, +6.9] | no |
| | DINOv2 | +52.2 | FAIL | 4/6 | $-0.2$ [$-5.9$, +5.7] | no |
| | PlantCLEF | $+12.4^{\dagger}$ | FAIL | 3/6 | $-7.2$ [$-13.4$, $-1.4$] | no |
| Oracle-normalized AUC | DINOv3 | +15.7 | PASS | 6/6 | +10.9 [+4.2, +18.7] | no |
| | DINOv2 | $+8.5^{\dagger}$ | PASS | 6/6 | +9.4 [+2.5, +14.5] | no |
| | PlantCLEF | $-10.2^{\dagger}$ | FAIL | 3/6 | $-5.2$ [$-10.7$, $-1.0$] | no |
| COVERED | | | | | | |
| Discovery AUC | DINOv3 | +39.1 | FAIL | 5/6 | +0.5 [$-0.4$, +1.2] | equiv. |
| | DINOv2 | +40.5 | FAIL | 5/6 | $-0.8$ [$-1.5$, $-0.1$] | no |
| | PlantCLEF | +45.6 | FAIL | 5/6 | $-0.5$ [$-1.0$, +0.1] | equiv. |
| Trapezoid AUC | DINOv3 | +39.8 | FAIL | 5/6 | +0.6 [$-0.4$, +1.2] | no |
| | DINOv2 | +41.5 | FAIL | 5/6 | $-0.6$ [$-1.3$, +0.1] | equiv. |
| | PlantCLEF | +47.1 | FAIL | 5/6 | $-0.6$ [$-1.1$, +0.0] | equiv. |
| Final $u_K$ | DINOv3 | +21.3 | PASS | 6/6 | +1.1 [+0.3, +1.9] | no |
| | DINOv2 | +21.5 | FAIL | 4/6 | $-0.3$ [$-1.0$, +0.5] | equiv. |
| | PlantCLEF | +23.1 | FAIL | 5/6 | +0.1 [$-0.1$, +0.4] | equiv. |
| Early AUC (1–3) | DINOv3 | +68.9 | FAIL | 5/6 | $-6.9$ [$-8.7$, $-5.1$] | no |
| | DINOv2 | +71.1 | FAIL | 5/6 | $-8.5$ [$-10.1$, $-6.7$] | no |
| | PlantCLEF | +76.1 | FAIL | 5/6 | $-3.1$ [$-4.5$, $-1.0$] | no |
| Oracle-normalized AUC | DINOv3 | +39.1 | FAIL | 5/6 | +0.5 [$-0.4$, +1.2] | equiv. |
| | DINOv2 | +40.5 | FAIL | 5/6 | $-0.8$ [$-1.5$, $-0.1$] | no |
| | PlantCLEF | +45.6 | FAIL | 5/6 | $-0.5$ [$-1.0$, +0.1] | equiv. |

drawn. CoreSet has the higher mean in $78.0\%$ of DINOv3 subsets, but in only $6.2\%$ and $10.4\%$ of DINOv2-Giant and PlantCLEF2024 subsets. The Holm rule never fires at five seeds, as the floor requires. Subsets of one sixteen-seed sample overlap. So these percentages describe resampling from the observed runs, not error rates over independent replications.

## N PER-ENCODER VALUES OF THE DRONE-POOL RESULTS

The main text gives DINOv3 values for the drone-pool results, and Table N.1 gives all three encoders. Every row is under REMOVED except the NAIVE, SELECTABLE and scikit-activeml rows. Those rows compare arms within one loop that keeps the unlabelable items (Appendices C.2 and O.1). The TypiClust and ProbCover rows come from the mechanism runs. The label-free ProbCover radius is the median nearest-neighbor distance on each encoder.

Table M.3: **Full specification table, continued** from Table M.2.

| | | vs. random | Joint, core | | CoreSet vs. NovelMargin | |
|---|---|---|---|---|---|---|
| Endpoint | Encoder | gain (%) | IUT | Holm | gain (%) [BCa 95% CI] | TOST |
| COVERED, novel-only | | | | | | |
| Discovery AUC | DINOv3 | +41.7 | PASS | 6/6 | +1.2 [+0.5, +1.8] | no |
| | DINOv2 | +43.3 | FAIL | 5/6 | −0.4 [−1.3, +0.4] | no |
| | PlantCLEF | +48.4 | FAIL | 5/6 | −0.2 [−0.8, +0.4] | equiv. |
| Trapezoid AUC | DINOv3 | +42.5 | PASS | 6/6 | +1.3 [+0.6, +1.9] | no |
| | DINOv2 | +44.4 | FAIL | 5/6 | −0.2 [−1.1, +0.7] | equiv. |
| | PlantCLEF | +50.1 | FAIL | 5/6 | −0.2 [−0.8, +0.5] | equiv. |
| Final $u_K$ | DINOv3 | +23.0 | PASS | 6/6 | +1.3 [+0.5, +2.1] | no |
| | DINOv2 | +23.4 | FAIL | 4/6 | +0.0 [−0.7, +0.8] | equiv. |
| | PlantCLEF | +24.7 | PASS | 6/6 | +0.5 [+0.2, +0.8] | equiv. |
| Early AUC (1–3) | DINOv3 | +69.9 | FAIL | 5/6 | −6.4 [−8.0, −4.2] | no |
| | DINOv2 | +73.3 | FAIL | 5/6 | −8.5 [−10.6, −6.5] | no |
| | PlantCLEF | +77.9 | FAIL | 5/6 | −4.3 [−6.1, −2.5] | no |
| Oracle-normalized AUC | DINOv3 | +41.7 | PASS | 6/6 | +1.2 [+0.5, +1.8] | no |
| | DINOv2 | +43.4 | FAIL | 5/6 | −0.4 [−1.3, +0.4] | no |
| | PlantCLEF | +48.4 | FAIL | 5/6 | −0.2 [−0.8, +0.4] | equiv. |
| REMOVED | | | | | | |
| Discovery AUC | DINOv3 | +46.3 | PASS | 6/6 | +5.5 [+4.5, +6.3] | no |
| | DINOv2 | +46.8 | PASS | 6/6 | +3.1 [+2.4, +3.9] | no |
| | PlantCLEF | +50.6 | PASS | 6/6 | +2.2 [+1.6, +2.7] | no |
| Trapezoid AUC | DINOv3 | +47.1 | PASS | 6/6 | +5.6 [+4.6, +6.4] | no |
| | DINOv2 | +47.6 | PASS | 6/6 | +3.1 [+2.4, +3.8] | no |
| | PlantCLEF | +52.2 | PASS | 6/6 | +2.2 [+1.6, +2.7] | no |
| Final $u_K$ | DINOv3 | +22.9 | PASS | 6/6 | +2.1 [+1.4, +2.8] | no |
| | DINOv2 | +24.4 | PASS | 6/6 | +2.6 [+1.8, +3.5] | no |
| | PlantCLEF | +23.7 | FAIL | 4/6 | +0.5 [+0.1, +0.9] | equiv. |
| Early AUC (1–3) | DINOv3 | +90.4 | PASS | 6/6 | +5.9 [+3.7, +7.5] | no |
| | DINOv2 | +89.5 | FAIL | 5/6 | +1.3 [−0.4, +3.3] | no |
| | PlantCLEF | +93.4 | PASS | 6/6 | +4.9 [+3.0, +7.0] | no |
| Oracle-normalized AUC | DINOv3 | +46.3 | PASS | 6/6 | +5.5 [+4.5, +6.3] | no |
| | DINOv2 | +46.8 | PASS | 6/6 | +3.1 [+2.4, +3.9] | no |
| | PlantCLEF | +50.6 | PASS | 6/6 | +2.2 [+1.6, +2.7] | no |
| All labeled | | | | | | |
| Discovery AUC | iNat slice | +39.6 | FAIL | 4/6 | −0.6 [−1.7, +0.7] | no |
| | PlantNet subset | +64.3 | PASS | 6/6 | +2.8 [+1.6, +4.1] | no |
| Trapezoid AUC | iNat slice | +39.8 | FAIL | 4/6 | −0.8 [−1.9, +0.5] | no |
| | PlantNet subset | +65.2 | PASS | 6/6 | +2.7 [+1.6, +4.0] | no |
| Final $u_K$ | iNat slice | +18.7 | FAIL | 3/6 | −1.3 [−3.0, +0.6] | no |
| | PlantNet subset | +26.6 | FAIL | 3/6 | +0.3 [−0.5, +1.3] | equiv. |
| Early AUC (1–3) | iNat slice | +95.3 | PASS | 6/6 | +3.7 [+1.1, +6.6] | no |
| | PlantNet subset | +140.8 | PASS | 6/6 | +6.7 [+2.3, +10.6] | no |
| Oracle-normalized AUC | iNat slice | +41.5 | FAIL | 4/6 | −0.3 [−1.4, +1.0] | no |
| | PlantNet subset | +64.3 | PASS | 6/6 | +2.8 [+1.6, +4.1] | no |

# O FAILED PICKS IN REAL TOOLS

## O.1 REPRODUCTION IN SCIKIT-ACTIVEML

`scripts/repro_skactiveml.py` runs scikit-activeml v1.0.0's own `pool.CoreSet` and `pool.RandomSampling` on the drone pool. It uses the paper's embeddings, rare-class defini-

Table M.4: **Power under REMOVED, discovery AUC.** Columns give CoreSet's gain in percent and the minimum detectable effect at 80% power, as a percent of the baseline mean. The MDE uses a noncentral $t$ at one-sided $\alpha = 0.025$ and $n = 16$. The last column is TOST power at a true difference of zero, with a margin of $\pm 2\%$ of random's mean.

| | DINOv3 | | | DINOv2 | | | PlantCLEF | | |
|---|---|---|---|---|---|---|---|---|---|
| CoreSet vs. | gain | MDE | TOST power | gain | MDE | TOST power | gain | MDE | TOST power |
| Random | +46.3 | 2.7 | 0.10 | +46.8 | 2.7 | 0.07 | +50.6 | 2.2 | 0.43 |
| $k$NN distance | +1.1 | 0.9 | 0.98 | +0.6 | 1.1 | 0.83 | +1.5 | 0.8 | 0.99 |
| NovelMargin | +5.5 | 1.3 | 0.73 | +3.1 | 1.2 | 0.85 | +2.2 | 0.9 | 0.98 |
| MixedHalf | +12.1 | 1.9 | 0.19 | +12.1 | 2.1 | 0.04 | +13.6 | 1.7 | 0.41 |
| LOF | +23.4 | 1.6 | 0.70 | +21.9 | 2.0 | 0.26 | +12.9 | 1.0 | 0.97 |
| Isolation Forest | +33.4 | 3.0 | 0.00 | +28.5 | 3.5 | 0.00 | +37.9 | 3.1 | 0.00 |
| BADGE (gradient) | +6.4 | 2.4 | 0.00 | +6.2 | 2.3 | 0.00 | +6.5 | 2.0 | 0.00 |
| MaxHerding | +9.5 | 1.0 | 0.96 | +10.0 | 1.3 | 0.80 | +6.3 | 1.4 | 0.56 |
| NNDB | +37.5 | 1.8 | 0.70 | +42.6 | 2.2 | 0.42 | +44.7 | 1.4 | 0.93 |
| ProbCover | +45.4 | 3.8 | 0.00 | +44.4 | 4.7 | 0.00 | +48.2 | 3.7 | 0.00 |
| $k$-means++ | +49.2 | 3.0 | 0.00 | +56.6 | 3.7 | 0.00 | +65.3 | 3.6 | 0.00 |
| TypiClust | +70.2 | 5.7 | 0.00 | +76.2 | 3.9 | 0.00 | +84.5 | 4.1 | 0.00 |

Table M.5: **Exact-test $p$-floor at $n$ seeds and the maximum simultaneous comparisons claimable at one-sided $\alpha = 0.025$ under Holm step-down.**

| $n$ | $p$-floor $= 1/2^n$ | Max $m$ for $p \le 0.025/m$ |
|---|---|---|
| 3 | 0.125 | 0 |
| 5 | 0.031 | 0 |
| 6 | 0.016 | 1 |
| 8 | 0.0039 | 6 |
| 10 | 0.00098 | 25 |
| 16 | $2^{-16}$ | 1638 |

tion and discovery AUC, and the embedding hashes match the evidence files. The runs use $K = 10$ rounds of $k = 200$ and seeds 0 to 15. In the library's default loop (`candidates=None`), the candidates are the rows whose label is still `NaN`. So a failed pick stays `NaN`, and the loop re-proposes it. In the second arm, the caller passes as candidates only rows never inspected. By round 10, default CoreSet re-proposes 197 to 199 of its 200 picks per round, and default random 30. Table O.1 gives the four arms. Default CoreSet falls below random on every encoder. Skipping inspected rows puts CoreSet above random on all 16 seeds. Its mean discovery AUC then lands within 2.1% of the SELECTABLE CoreSet runs of Appendix C.2.

Three caveats apply. (i) The run is a cold start with no labels. The paper's 200-row initial pool is a convention of our simulator, not of the library's loop. With no labeled rows, CoreSet has no centers, every candidate has the same utility, and its first pick is a tie-break. (ii) Here an unlabelable item is a row with no ground-truth label. As under NAIVE, the 4,437 never-labeled rows play the part of failed picks. (iii) Neither random arm uses the embedding, so its values repeat across encoders.

### O.2 CODE AUDIT OF WHAT TOOLS DO WITH FAILED PICKS

We read the pool-update code of 19 tools: active-learning libraries, research repositories and one annotation tool. Each is pinned to a release tag or commit. `data/evidence/code_audit/audit.json` gives file and line references. For each tool, we recorded what happens to a failed pick on the next round (Table O.2). *Re-proposed*: the pick stays a candidate. *Removed*: it leaves the pool. *Known out-of-distribution (OOD) set*: it leaves the pool and joins a separate set of out-of-distribution picks. *Forced label*: the explicit no-label call raises an error. *Cannot occur*: the shipped oracle always holds the true label. *Not applicable*: the tool has no open multi-round loop to audit. The open-set repositories LfOSA and EOAL share a lineage and near-identical sampling signatures, but they handle failed picks differently. LfOSA

Table M.6: **Five-seed replay of the sixteen-seed COVERED evidence.** Percentage of the 4,368 five-seed subsets in which each decision rule favors an arm. "Holm reject" is the sixteen-seed decision at one-sided $\alpha = 0.025$ over the six-alternative family. It uses SciPy Wilcoxon, and the exact sign-flip test gives the same marks. No five-seed subset gives a Holm rejection.

| | | 16 seeds | five-seed subsets (%) | | |
|---|---|---|---|---|---|
| Encoder | CoreSet vs. | Holm reject | CoreSet higher mean | $p < 0.05$, CoreSet | $p < 0.05$, alternative |
| DINOv3 | MixedHalf | ✓ | 100.0 | 100.0 | 0.0 |
| | MaxHerding | ✓ | 100.0 | 68.8 | 0.0 |
| | NovelMargin | × | 78.0 | 10.6 | 0.0 |
| | ProbCover | ✓ | 100.0 | 100.0 | 0.0 |
| | TypiClust | ✓ | 100.0 | 100.0 | 0.0 |
| | $k$-means++ | ✓ | 100.0 | 100.0 | 0.0 |
| DINOv2 | MixedHalf | ✓ | 100.0 | 100.0 | 0.0 |
| | MaxHerding | ✓ | 100.0 | 100.0 | 0.0 |
| | NovelMargin | × | 6.2 | 0.1 | 5.8 |
| | ProbCover | ✓ | 100.0 | 100.0 | 0.0 |
| | TypiClust | ✓ | 100.0 | 100.0 | 0.0 |
| | $k$-means++ | ✓ | 100.0 | 100.0 | 0.0 |
| PlantCLEF | MixedHalf | ✓ | 100.0 | 100.0 | 0.0 |
| | MaxHerding | ✓ | 100.0 | 100.0 | 0.0 |
| | NovelMargin | × | 10.4 | 0.0 | 18.1 |
| | ProbCover | ✓ | 100.0 | 100.0 | 0.0 |
| | TypiClust | ✓ | 100.0 | 100.0 | 0.0 |
| | $k$-means++ | ✓ | 100.0 | 100.0 | 0.0 |

Table N.1: **Per-encoder gains on the drone pool**, discovery AUC, $n_s = 16$, percent. "$A$ vs. $B$" is $A$'s gain over $B$. The two indented rows modify TypiClust cumulatively. The first two SELECTABLE rows coincide on DINOv3 and DINOv2 because both CoreSet runs sum to the same discovery AUC over the 16 seeds. Their per-seed values differ.

| Comparison | DINOv3 | DINOv2 | PlantCLEF |
|---|---|---|---|
| CoreSet vs. random | +46.26 | +46.81 | +50.60 |
| CoreSet vs. NovelMargin | +5.49 | +3.12 | +2.19 |
| CoreSet vs. label-free ProbCover | +22.64 | +23.18 | +20.89 |
| TypiClust vs. random | −14.09 | −16.66 | −18.35 |
| without the small-cluster filter | +10.57 | +5.98 | +10.61 |
| and picking the least typical point | +30.02 | +31.58 | +36.23 |
| ProbCover, purity radius, vs. random | +0.60 | +1.67 | +1.60 |
| ProbCover, label-free radius, vs. random | +19.27 | +19.18 | +24.58 |
| NAIVE CoreSet vs. random | −27.23 | −32.51 | −49.51 |
| NAIVE $k$NN distance vs. random | −27.89 | −22.09 | −56.22 |
| NAIVE NovelMargin vs. random | −20.18 | −23.90 | −40.20 |
| SELECTABLE CoreSet vs. NAIVE random | +71.17 | +71.17 | +54.74 |
| SELECTABLE CoreSet vs. $\text{Random}_{\text{xf}}$ | +50.72 | +50.72 | +36.26 |
| SELECTABLE CoreSet vs. CoreSet under REMOVED | −46.89 | −47.09 | −53.37 |
| scikit-activeml CoreSet vs. random, default candidates | −19.74 | −28.04 | −36.06 |
| scikit-activeml CoreSet vs. random, uninspected candidates | +46.04 | +44.17 | +32.62 |

subtracts only the known-class picks from the unlabeled pool (`AL_center.py:164`), so it can select its unknown-class picks again. EOAL makes the same subtraction, but it also collects the unknown-class picks in `invalidList` (`main.py:156-158`). It then builds the next round's candidate pool without them (`main.py:165`). The audit has six limits. The scikit-activeml and modAL entries come from a separate, prior check. modAL keeps no pool state, and its removal is the documented example's `np.delete`. baal re-proposes a row the caller never passes to `label()`, and it raises an error only when a label is required but missing. cleanlab ActiveLab is a stateless scoring function, so its behaviour depends on the caller. Lightly's selection code is no

Table O.1: **scikit-activeml v1.0.0, cold start, BCI**, mean discovery AUC over 16 seeds. *Default*: the library's default candidates, so the loop re-proposes a failed pick. *Skipped*: the loop skips rows already inspected. Gain is CoreSet over random within a loop, with the exact one-sided sign-flip $p$ in the paper's direction. Seeds is the number of seeds on which CoreSet is ahead. The last column is SELECTABLE CoreSet in the paper's simulator (Appendix C.2).

| | Default loop | | | | Inspected rows skipped | | | | Paper, SELECTABLE | |
|---|---|---|---|---|---|---|---|---|---|---|
| Encoder | CoreSet | Random | Gain, $p$ | Seeds | CoreSet | Random | Gain, $p$ | Seeds | CoreSet | Diff. |
| DINOv3 | 314.81 | 392.25 | $-19.7\%$, 1.000 | 1/16 | 590.62 | 404.44 | $+46.0\%$, $2^{-16}$ | 16/16 | 595.25 | $-0.78\%$ |
| DINOv2 | 282.25 | 392.25 | $-28.0\%$, 1.000 | 0/16 | 583.06 | 404.44 | $+44.2\%$, $2^{-16}$ | 16/16 | 595.25 | $-2.05\%$ |
| PlantCLEF | 250.81 | 392.25 | $-36.1\%$, 1.000 | 0/16 | 536.38 | 404.44 | $+32.6\%$, $2^{-16}$ | 16/16 | 538.12 | $-0.33\%$ |

longer open source. Label Studio's entry is its default skip queue, which returns a skipped task to other annotators.

Table O.2: **What happens to a failed pick on the next round**, by tool, pinned to the release tag or commit we read. Short SHAs are in typewriter.

| Tool | Kind | Version / commit | Next round |
|---|---|---|---|
| scikit-activeml | library | 1.0.0 (`2281691`) | re-proposed |
| libact | library | v0.1.6 (`ade1e4b`) | re-proposed |
| LfOSA | open-set research code | `3e22f75` | re-proposed |
| Label Studio | annotation tool | `a77fe7f` | re-proposed |
| modAL | library | 0.4.2.1 | removed |
| small-text | library | v1.4.1 (`c8a19ba`) | removed |
| CCAL | open-set research code | `76ee1dd` | known-OOD set |
| MQNet | open-set research code | `77b4b5c` | known-OOD set |
| EOAL | open-set research code | `baadd13` | known-OOD set |
| baal | library | v2.1.0 (`0e7e0c3`) | forced label |
| ALiPy | library | v1.2.5 (`bc69062`) | cannot occur |
| DeepAL | research code | `eea3d2c` | cannot occur |
| DISTIL | library | `d46f0b4` | cannot occur |
| TypiClust | research code | `4097a71` | cannot occur |
| ProbCover | research code | `4097a71` | cannot occur |
| USL | cold-start research code | `ceaab3c` | not applicable |
| ActiveFT | cold-start research code | `0687b00` | not applicable |
| Lightly | library | `3684a18` | not applicable |
| cleanlab ActiveLab | library | v2.9.0 (`7506257`) | unclear |

## P EXPLORATORY EMBEDDING-GEOMETRY ANALYSES

These analyses are exploratory, chosen after the main results, and use COVERED on the drone pool. The iNaturalist and PlantNet inputs lack complete hash-linked records of the subset and the encoder extraction (Appendix Q). So the associations below describe the stored artifacts and do not show that they generalize to the source populations. Transform rungs from one dataset are dependent observations, so the pooled fit is descriptive.

Separability is how often a labeled image's nearest neighbor in the embedding has the same class. This is the $k$NN leave-one-out accuracy, kNN-LOO. We keep a dataset and its labels fixed and vary only the geometry through a ladder of label-free transforms. The transforms are additive noise, random projection and PCA-whitening. At each rung we re-audit the gain of CoreSet over random ($n_s = 16$). The transforms change more than separability, so the ladder does not isolate separability as a mediator. Gain rises with induced separability on the drone pool, PlantNet-300K and an iNaturalist slice, with within-dataset Spearman $\rho = 0.76$, $0.94$ and $0.96$. Pooling the rungs below $80\%$ separability gives $\text{gain}\% = 0.66 \cdot \text{sep}\% + 6.0$, with slope $95\%$ CI $[0.51, 0.80]$, $R^2 = 0.79$ and $n = 25$ rungs. The slope is $0.67$ on the drone pool alone and $0.73$ on PlantNet alone. Above about $80\%$ separability the benefit plateaus. On iNaturalist, gain at separability $\geq 85\%$ is about

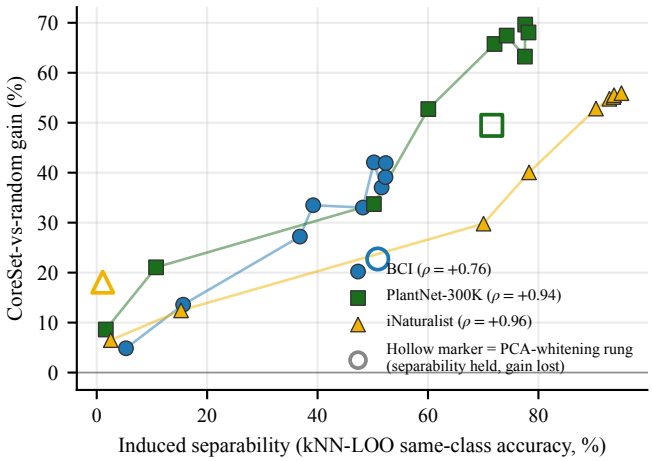



Figure P.1: **Embedding geometry and discovery gain: one label-free transform ladder per dataset.** Each dataset and its labels stay fixed. We blur only the embedding geometry through the same ladder of additive noise, random projection and PCA-whitening. At each rung we re-audit the gain of CoreSet over random ($n_s = 16$). Gain rises with induced separability on the three tested datasets, with within-dataset Spearman $\rho = 0.76$, $0.94$ and $0.96$. The hollow marker is the PCA-whitening rung. On the drone pool it holds $k$NN-LOO separability almost fixed but loses $\sim 16$ points of gain.

$+55\%$, against a straight-line prediction near $+67\%$. PCA-whitening is a counterexample. It holds separability almost fixed ($52.3\%$ to $50.9\%$) but costs 16 points of gain. It also raises the intrinsic dimensionality, with an MLE estimate going from 9 to 35. BioCLIP-2 has the lowest separability of the tested backbones on aerial drone-pool crowns, with $\sim 7\%$ separability and negative gain. It is among the best on ground-level plant photos, with 82 to $99\%$ separability and $+54$ to $+79\%$ gain.

**Exploratory audio extension.** On a long-tailed bird-vocalization dataset, gain rises with induced separability as on the image datasets (Figure P.2). The dataset is CBI, with 1552 clips, 111 classes, 12 rare classes and Perch v2 embeddings. Spearman $\rho$ is $0.94$ over all ten rungs. With the whitening rung set aside, $\rho = 0.97$, $p < 10^{-4}$ and the bootstrap $95\%$ CI is $[0.74, 1.00]$. The relationship survives a different seed base, budget and rare threshold, with $\rho = 0.93$ to $0.97$ when the whitening rung is set aside. A balanced audio control, HumBug, has no rare classes. So its rare-class discovery is zero by definition, and it cannot test relative benefit. The local audio slope is $1.25$ at $k = 40$ per round and $0.53$ at $k = 60$. So on audio we claim the sign and monotonicity of the relationship, not the image-fitted slope. A larger slice of the same corpus has 261 classes, 2500 clips and 87 rare classes. Cutting its class count to 31 with the ladder fixed flips the Spearman from $+0.90$ to $-0.10$. A 31-class marine-mammal audio set and 20-class news text show negative relationships (audio $\rho = -0.6$, with an inverted U; text $\rho = -0.5$).

**A pre-flight diagnostic.** The package offers an exploratory diagnostic, `predict_al_benefit`. It maps $k$NN-LOO separability on a candidate embedding to an expected CoreSet gain. It gives an estimate only inside its calibration range, which is at least 30 classes and separability between about $7\%$ and $60\%$. Outside that range it declines to estimate. It is calibrated across 22 drone-pool backbones, so it spans the gap between high- and low-separability families. As a result, its slope of $1.28$ points of gain per point of separability is twice the within-dataset slope of $0.66$. No backbone falls between $24\%$ and $50\%$ separability, so the diagnostic interpolates that band and does not measure it. It is not a validated decision rule for new datasets. It can mispredict across encoders. On D4, PlantCLEF lowers rare-class separability but moves CoreSet's gain over random from $-25.7\%$ to $+17.9\%$ (Appendix T).

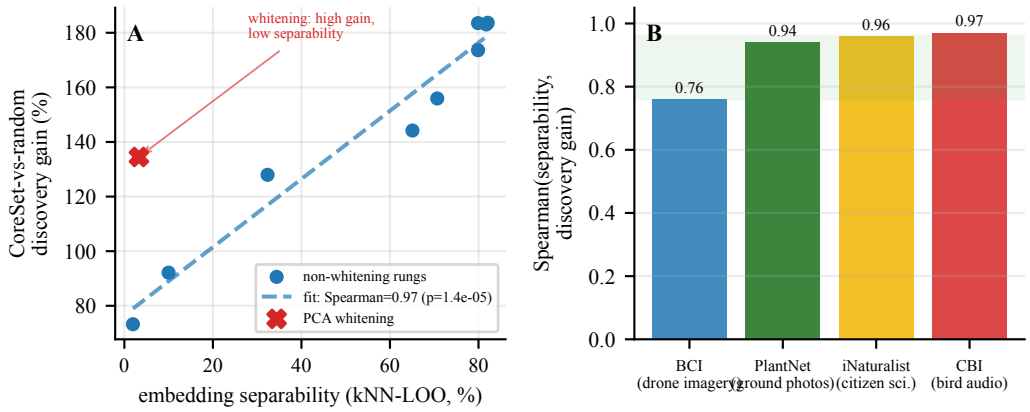



Figure P.2: **An exploratory separability ladder on audio.** *(A)* On a long-tailed bird-vocalization dataset (CBI, Perch v2 embeddings), the same label-free transform ladder changes both the gain of CoreSet over random and induced separability. Spearman $\rho$ is 0.97 without the whitening rung and 0.94 with it. The PCA-whitening rung (red) sits off the line. *(B)* Spearman(separability, gain) across modalities. The three image datasets (green) and long-tailed audio (blue) show positive associations across the tested ladders, except for whitening. The balanced audio control (HumBug) has no rare classes to discover.

## Q Exploratory additional datasets, gradient BADGE, and downstream accuracy

This appendix reports exploratory audits chosen after we saw the COVERED main result. They use $n_s = 16$ seeds, $K = 10$ rounds and an initial pool of 200.

**Datasets.** We reuse the PlantNet-300K subset and the iNaturalist slice of Appendix P. The PlantNet subset has 4,062 images, 200 species and 26 rare species. The iNaturalist slice has 825 images, 100 species and 35 rare species. The local artifacts identify both embeddings as DINOv3 ViT-L/16, but hash-linked checkpoint and preprocessing records are unavailable. Every row is labeled, so the discovery protocol has no unlabelable items and the three pool conventions coincide. The iNaturalist tail comes from sampling 5 to 10 images per species, so its 35 rare species are rare by sampling. The PlantNet subset's selection rule, seed and source split are not recorded. The per-round budget is $k = 105$ on PlantNet and $k = 21$ on iNaturalist. In the statistics audit, the core family's joint claim passes on PlantNet and fails on iNaturalist (Table M.2).

**Gradient BADGE.** Each round, a multinomial logistic head is fit on the rows labeled so far. Every pool row gets its last-layer gradient embedding under its own predicted label. The batch is drawn by $k$-means++ in that space (Ash et al., 2020). On PlantNet, a rerun at the pinned commit replaced its trajectories in the audit statistics; Table Q.1 keeps the original trajectories. The rerun changed 15 of 16 seeds and moved its mean discovery AUC from 171.0 to 168.0. On iNaturalist, 2 seeds changed. The committed files have no provenance record and were likely produced under scikit-learn 1.8.0.

**Downstream protocol.** A fixed stratified 20% of the labeled rows is the test set. It enters the simulator as covered rows, so no selector can pick it. After each round, a logistic head is fit on the rows labeled so far and scored on the test set. The scores are overall accuracy, class-balanced accuracy, and accuracy on test rows of rare classes. A class not yet discovered is scored wrong. The endpoint is the mean over the $K$ rounds. Test embeddings stay visible while their labels are withheld, so this is a transductive evaluation. No accuracy endpoint was specified in advance, and these endpoints are not corrected across metrics, datasets or encoders.

**Results.** The discovery gain does not consistently carry over to accuracy (Table Q.1). The changes are relative percentages, not percentage points. CoreSet's mean test accuracy against random is

$-0.96\%$, $-1.41\%$ and $+0.38\%$ on the drone-pool encoders, $+1.18\%$ on iNaturalist and $-9.47\%$ on PlantNet. NovelMargin and gradient BADGE lead random on class-balanced accuracy on all five rows. Both use the labels, and CoreSet does not. Finding a rare class once is not the same as learning to recognize it. On the drone pool, a rare class has at most five examples, and some of them are in the test set.

**Mixing selectors.** A post hoc sweep under the downstream protocol combines CoreSet with NovelMargin or gradient BADGE. A mixed selector either switches after $s \in \{1, 2, 3, 5, 7\}$ rounds or draws a fraction $\rho \in \{0.25, 0.5, 0.75\}$ of each batch from CoreSet: 16 arms per BCI encoder. Endpoints rerun in this environment reproduce the committed discovery trajectories exactly, and seed-mean accuracies shift by under $0.1$ points. No arm beats both CoreSet and its label-aware endpoint on discovery AUC and class-balanced accuracy with all four paired seed-bootstrap $95\%$ intervals above zero. The closest, $\rho = 0.25$ with NovelMargin on PlantCLEF, gains $+3.9\,[-1.5, +8.9]$ AUC over CoreSet. Adding gradient BADGE beats CoreSet on both axes in 17 of 24 arms but trails pure gradient BADGE by $0.3$ to $2.3$ points of class-balanced accuracy. NovelMargin's AUC is within $1\%$ of CoreSet's ($+0.9\%$, $+0.6\%$, $-0.1\%$ on DINOv3, DINOv2, PlantCLEF).

**CIFAR-100-LT convention check.** This exploratory check asks whether the ranking of CoreSet and random under each pool convention carries over to a non-ecological pool. We chose it after the ecology results, but committed its design (`3e0d3f0`) before any run. CIFAR-100-LT with imbalance 100 gives a pool of 10,847 images. 52 classes, $60.3\%$ of the pool, are unlabelable, close to BCI's $57.5\%$. The 30 labelable classes with at most 56 pool images are rare, which matches the drone pool's rare share. We use 16 seeds, $K = 10$ rounds of $k = 200$ and exact one-sided paired sign-flip tests on the AUC differences, uncorrected. We dropped TypiClust for runtime, a logged deviation outside the contrast. Random's picks do not depend on the encoder. With DINOv2 ViT-g/14, CoreSet leads random under all three pool conventions. It leads under REMOVED (296.3 vs 263.8, ahead on 16 of 16 seeds) and NAIVE (186.6 vs 134.7, 15 of 16). Under SELECTABLE, $\text{CoreSet}_{\text{xf}}$ leads $\text{Random}_{\text{xf}}$ (251.8 vs 221.3, 15 of 16). With CLIP ViT-L/14, CoreSet leads only under REMOVED (279.9 vs 263.8, 16 of 16). Random leads under NAIVE (134.7 vs 104.1, CoreSet ahead on 5 of 16, $p = 0.0019$). $\text{Random}_{\text{xf}}$ leads $\text{CoreSet}_{\text{xf}}$ (221.3 vs 203.9, 2 of 16, $p = 1.4 \times 10^{-4}$). A linear probe on CoreSet's round-10 labels under REMOVED reaches top-1 accuracy $0.598$ with CLIP and $0.890$ with DINOv2. So whether seeking distance pays off varies with the encoder, as on D4 (Appendix T). On CIFAR, random wins under the encoder with the lower linear-probe accuracy, CLIP. On D4, distance wins under the encoder with the lower rare-class separability, PlantCLEF. So separability alone does not predict the sign. The check is exploratory. It bounds the scope of H2b, since $\text{CoreSet}_{\text{xf}}$ does not beat $\text{Random}_{\text{xf}}$ on every pool and encoder.

**Second encoder on D1 to D4.** This exploratory rerun repeats the full preregistered arm set on D1 to D4 with CLIP ViT-L/14 image embeddings in place of DINOv3. We made it after the confirmatory verdicts. Manifests, rare classes, budgets, seeds and decision rules are unchanged, and the rerun carries no confirmatory weight. It gives the same three verdicts. H2's pattern repeats on both datasets. $\text{CoreSet}_{\text{xf}}$ beats $\text{Random}_{\text{xf}}$ by $+197.7\%$ on D1 and $+732.2\%$ on D2. $\text{CoreSet}_{\text{af}}$ beats $\text{CoreSet}_{\text{xf}}$ by $+16.1\%$ and $+57.7\%$. All four tests have exact sign-flip $p = 2^{-16}$. H1 again fails. Distance beats density on D1 to D3 but not on D4 ($-12.1\%$), and the meta-analysis gives $p = 0.042$. H3 again fails, since CoreSet still leads random under NAIVE on both datasets. Unlike on CIFAR-100-LT, CLIP does not reverse H2b on D1 and D2. So the CIFAR reversal is not a property of CLIP alone. The evidence is in `data/evidence/exploratory/clip/`.

## R STOPPING CRITERION

We fit the cumulative rare-class discovery curve to an exponential saturation model $N \cdot (1 - e^{-r/\tau})$. We also track the marginal discovery rate over a sliding window. Figure R.1 shows both on a synthetic long-tail dataset ($n = 5{,}000$, 100 classes, Zipf $s = 1.5$, 38 rare classes). The fit estimates $\hat{N} = 38$ reachable rare classes. The marginal rate drops below $0.5$ per round at round 11, when $92\%$ of reachable rare classes are found. On the drone pool, `analyze_stopping` with its default flags fires on the saturation target at rounds 22–23. It estimates 154–159 rare classes, against 157 in the full labeled pool. That run uses DINOv3, CoreSet, 40 rounds of 80, seeds 0–2, an initial pool of 200 and rare threshold 5 (`scripts/stopping_on_real_runs.py`). The nominal

Table Q.1: **Audits on other datasets.** Percentage gain over random (Appendix W) with its bootstrap 95% interval ($n_s = 16$). Rare-class AUC is under the discovery protocol (COVERED on the drone pool). The three accuracy columns are under the downstream protocol and are exploratory.

| Strategy | Rare-class AUC | Accuracy | Balanced acc. | Rare-class acc. |
|---|---|---|---|---|
| *iNaturalist slice* ($k = 21$) | | | | |
| CoreSet | +39.6 [+33.3, +46.6] | +1.2 [+0.1, +2.4] | +2.5 [+1.4, +3.8] | +10.4 [+7.1, +13.9] |
| NovelMargin | +40.5 [+34.6, +47.3] | +6.5 [+5.5, +7.6] | +9.1 [+7.7, +10.7] | +24.5 [+19.6, +30.0] |
| MaxHerding | +38.6 [+32.7, +45.4] | +3.5 [+2.5, +4.8] | +5.3 [+4.1, +6.7] | +15.8 [+12.4, +19.4] |
| MixedHalf | +26.1 [+19.7, +33.2] | +1.3 [+0.2, +2.4] | +2.0 [+0.5, +3.5] | +6.0 [+1.6, +10.5] |
| BADGE (gradient) | +30.1 [+23.6, +36.6] | +5.1 [+4.2, +6.1] | +6.5 [+5.1, +8.0] | +14.8 [+9.7, +20.2] |
| $k$-means++ | −45.5 [−50.1, −40.9] | +0.1 [−0.9, +1.1] | −1.6 [−2.7, −0.4] | −11.4 [−14.4, −8.4] |
| ProbCover | +25.5 [+19.4, +32.4] | +3.1 [+2.0, +4.2] | +4.5 [+3.2, +5.9] | +12.6 [+8.8, +16.7] |
| TypiClust | −82.8 [−85.1, −80.2] | −3.3 [−4.7, −2.0] | −6.3 [−8.0, −4.9] | −23.9 [−28.1, −19.8] |
| *PlantNet-300K subset* ($k = 105$) | | | | |
| CoreSet | +64.3 [+57.8, +70.9] | −9.5 [−10.3, −8.6] | −8.0 [−9.1, −6.9] | +29.8 [+14.2, +49.4] |
| NovelMargin | +59.8 [+53.5, +66.1] | +0.1 [−0.7, +0.9] | +7.2 [+6.4, +8.1] | +112.2 [+88.0, +142.4] |
| MaxHerding | +48.4 [+42.4, +54.9] | −2.6 [−3.3, −2.0] | +0.6 [−0.1, +1.3] | +47.8 [+32.0, +67.4] |
| MixedHalf | +49.7 [+44.1, +55.6] | −3.4 [−4.1, −2.6] | −3.1 [−4.1, −2.2] | +15.6 [+1.6, +32.5] |
| BADGE (gradient) | +40.1 [+32.5, +47.9] | +4.8 [+4.1, +5.8] | +8.6 [+7.7, +9.8] | +65.9 [+47.6, +90.1] |
| $k$-means++ | −48.0 [−52.5, −43.5] | +0.3 [−0.4, +1.1] | −1.6 [−2.8, −0.5] | −24.1 [−38.1, −9.6] |
| ProbCover | −13.7 [−20.4, −6.7] | +0.9 [+0.2, +1.7] | +1.3 [+0.4, +2.1] | +24.0 [+12.8, +37.0] |
| TypiClust | −55.8 [−59.8, −51.8] | −0.4 [−1.2, +0.5] | 0.0 [−1.1, +1.2] | −34.9 [−43.2, −24.0] |
| *BCI, DINOv3* ($k = 200$) | | | | |
| CoreSet | +39.1 [+37.0, +41.3] | −1.0 [−1.5, −0.4] | −0.4 [−2.6, +2.0] | −2.6 [−23.2, +27.5] |
| NovelMargin | +38.4 [+36.2, +40.7] | +1.4 [+0.7, +2.0] | +7.7 [+5.4, +10.0] | +24.4 [−3.0, +66.0] |
| MaxHerding | +37.0 [+34.8, +39.4] | −1.0 [−1.6, −0.5] | 0.0 [−2.1, +2.3] | +11.1 [−10.7, +43.9] |
| MixedHalf | +25.5 [+22.8, +28.1] | −0.2 [−0.6, +0.2] | +0.9 [−0.5, +2.4] | +15.0 [−6.5, +49.1] |
| BADGE (gradient) | +35.7 [+33.9, +37.7] | +3.0 [+2.5, +3.4] | +8.3 [+6.1, +10.6] | +30.0 [+4.1, +67.2] |
| $k$-means++ | −1.9 [−4.2, +0.3] | +1.9 [+1.2, +2.7] | +1.9 [−0.5, +4.5] | −1.0 [−27.2, +35.7] |
| ProbCover | +4.3 [+1.8, +6.7] | +0.1 [−0.6, +0.7] | +0.1 [−1.9, +2.2] | +1.0 [−17.8, +27.2] |
| TypiClust | +4.9 [+2.0, +7.6] | +1.4 [+0.8, +2.0] | +4.9 [+2.6, +7.2] | +49.5 [+24.5, +87.7] |
| *BCI, DINOv2-Giant* ($k = 200$) | | | | |
| CoreSet | +40.5 [+38.4, +42.7] | −1.4 [−1.9, −0.9] | +0.6 [−1.3, +2.6] | 0.0 [−10.7, +14.6] |
| NovelMargin | +41.7 [+39.5, +44.2] | +1.5 [+1.0, +2.0] | +11.3 [+10.0, +12.6] | +53.6 [+40.6, +69.5] |
| MaxHerding | +36.2 [+34.1, +38.3] | −0.8 [−1.4, −0.2] | +0.5 [−1.4, +2.7] | −3.7 [−14.5, +10.6] |
| MixedHalf | +23.8 [+21.9, +25.8] | +0.1 [−0.6, +0.7] | +1.7 [−1.0, +4.7] | +8.2 [−9.4, +29.4] |
| BADGE (gradient) | +37.6 [+35.0, +40.0] | +3.4 [+2.9, +3.9] | +10.7 [+8.9, +12.5] | +24.2 [+8.7, +41.7] |
| $k$-means++ | −6.2 [−8.6, −3.9] | +1.6 [+1.1, +2.2] | +2.9 [+1.1, +4.9] | +15.7 [+1.9, +33.5] |
| ProbCover | +2.6 [−0.2, +5.5] | −0.3 [−1.1, +0.4] | +0.5 [−1.7, +2.5] | −3.0 [−16.6, +13.1] |
| TypiClust | +5.2 [+3.1, +6.9] | +0.2 [−0.3, +0.8] | +3.4 [+0.9, +5.7] | +33.6 [+16.6, +53.7] |
| *BCI, PlantCLEF2024* ($k = 200$) | | | | |
| CoreSet | +45.6 [+43.5, +47.8] | +0.4 [0.0, +0.8] | +3.3 [+1.5, +5.1] | +1.5 [−10.5, +15.0] |
| NovelMargin | +46.4 [+44.3, +48.6] | +2.5 [+2.1, +2.9] | +12.4 [+10.4, +14.3] | +51.3 [+34.8, +71.3] |
| MaxHerding | +42.3 [+40.2, +44.5] | +0.5 [+0.1, +0.9] | +3.9 [+2.0, +5.8] | +7.5 [−4.6, +20.7] |
| MixedHalf | +29.5 [+27.2, +31.9] | +0.7 [+0.2, +1.2] | +2.0 [−0.6, +4.5] | −3.3 [−19.0, +14.3] |
| BADGE (gradient) | +40.9 [+38.3, +43.5] | +3.7 [+3.0, +4.4] | +12.5 [+10.2, +14.8] | +38.8 [+24.1, +56.5] |
| $k$-means++ | −8.9 [−10.7, −6.9] | +1.4 [+0.8, +2.0] | +2.3 [+0.4, +4.1] | −21.8 [−31.1, −11.8] |
| ProbCover | +4.8 [+2.5, +7.1] | +0.4 [−0.3, +1.1] | +1.6 [−1.0, +4.2] | +4.0 [−13.8, +23.4] |
| TypiClust | +9.6 [+6.3, +13.0] | +1.6 [+1.1, +2.2] | +6.8 [+4.5, +9.1] | +32.4 [+17.2, +49.3] |

40-round budget exceeds the remaining eligible pool, so it is not a basis for a realized cost saving. Kurinchi-Vendhan & Beery (2026) propose a related rule from rarefaction curves. Ecology offers nonparametric alternatives such as Chao1 (Chao, 1984) and coverage-based rarefaction (Chao & Jost, 2012).

## S PREREGISTERED SECONDARY ENDPOINTS

The protocol names two secondary endpoints: the final count $u_K$ and the early AUC $u_1 + u_2 + u_3$. We report them with the same tests as the discovery AUC, and they carry no confirmatory weight. Table S.1 gives them for every component of Table 1. H2's pattern repeats on both endpoints: all 8 of its 8 components reject after Holm. H1 misses $\alpha$ on both, as on the discovery AUC. H3 under NAIVE again finds CoreSet ahead of random. On the discovery AUC, TOST of $\text{CoreSet}_{\text{af}}$ against $\text{CoreSet}_{\text{xf}}$ with a $\pm 2\%$ margin is inconclusive on D1 and inconclusive on D2. $\text{CoreSet}_{\text{af}}$'s lead exceeds the margin, so equivalence cannot be shown.

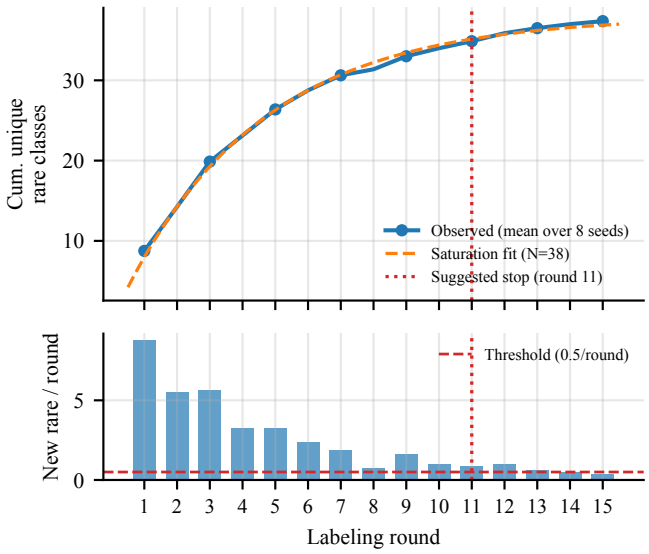



Figure R.1: **Stopping criterion on a synthetic long-tail dataset. Top:** cumulative rare-class discovery (blue) with exponential saturation fit (orange). **Bottom:** marginal discovery rate per round. The red line marks the suggested stopping round.

Table S.1: **Secondary endpoints of the preregistered tests.** Same components, tests and effect definition as Table 1, with $d$ computed on $u_K$ or on $u_1 + u_2 + u_3$ plus one. "met" means the primary decision rule is met on this endpoint. These endpoints do not change any verdict.

| | | Final $u_K$ | | | Early AUC $u_1 + u_2 + u_3$ | | |
|---|---|---|---|---|---|---|---|
| Hyp. | Dataset | Effect | $p$ | Result | Effect | $p$ | Result |
| H1 | D1 iWildCam | +122.5% | $2^{-16}$ | — | +325.2% | $2^{-16}$ | — |
| | D2 Serengeti | +95.9% | $2^{-16}$ | — | +367.4% | $2^{-16}$ | — |
| | D3 PlantNet | +143.8% | $2^{-16}$ | — | +521.8% | $2^{-16}$ | — |
| | D4 Nepenthaceae | −11.2% | 1.000 | — | −15.8% | 0.999 | — |
| | meta | +75.3% | 0.047 | not met | +219.3% | 0.041 | not met |
| H2a | D1 iWildCam | +5.7% | $2^{-16}$ | reject | +12.5% | $2^{-16}$ | reject |
| | D2 Serengeti | +31.4% | $3.1 \times 10^{-5}$ | reject | +10.3% | $1.8 \times 10^{-4}$ | reject |
| H2b | D1 iWildCam | +147.7% | $2^{-16}$ | reject | +444.4% | $2^{-16}$ | reject |
| | D2 Serengeti | +235.4% | $2^{-16}$ | reject | +889.5% | $2^{-16}$ | reject |
| | all | | | met | | | met |
| H3, REMOVED | D1 iWildCam | +72.0% | $2^{-16}$ | reject | +278.4% | $2^{-16}$ | reject |
| | D2 Serengeti | +33.5% | $6.1 \times 10^{-5}$ | reject | +363.6% | $2^{-16}$ | reject |
| H3, NAIVE | D1 iWildCam | +119.6% | 1.000 | retain | +390.1% | 1.000 | retain |
| | D2 Serengeti | +139.1% | 1.000 | retain | +636.4% | 1.000 | retain |
| | all | | | not met | | | not met |

**Secondary families.** The protocol also names one family per pool convention and dataset. Under SELECTABLE, the family compares $\text{CoreSet}_{\text{af}}$ against each of the other 15 arms. Under REMOVED, it compares CoreSet against each of the other 15 arms. Each comparison is the one-sided exact sign-flip test on the discovery AUC at $\alpha = 0.025$, with Holm step-down within the family. These families carry no confirmatory weight (disclosure (x) in Appendix X). Table S.2 gives every effect. Under SELECTABLE, $\text{CoreSet}_{\text{af}}$ beats 14 of 15 arms on D1 and 13 on D2. Plain CoreSet has $\text{CoreSet}_{\text{af}}$'s AUC on every seed. On D2, LOF is ahead of $\text{CoreSet}_{\text{af}}$ in point estimate, with $\text{CoreSet}_{\text{af}}$'s effect at −17.3%. Under REMOVED, CoreSet beats 12, 9, 11 and 3 of 15 arms on D1 to D4. The two arms with CoreSet's AUC on every seed cannot reject. No arm is ahead of CoreSet on D1. On D2,

BADGE ($-9.4\%$) and LOF ($-5.8\%$) are ahead. On D3, $k$NN distance is ahead ($-4.0\%$). On D4, 10 arms are ahead, including random ($-25.7\%$) and BADGE ($-61.5\%$). There CoreSet beats only $k$NN distance, NovelMargin and Isolation Forest. The reverse tests, of an arm beating the reference, were not preregistered and were not run. So these leads are point estimates only. These results do not make $\text{CoreSet}_{\text{af}}$ the best selector overall. H2 claims only that $\text{CoreSet}_{\text{af}}$ beats $\text{CoreSet}_{\text{xf}}$ among selectors that skip failed picks.

Table S.2: **Preregistered secondary families**, discovery AUC, $n_s = 16$. Each cell is the reference's effect over that arm, $\exp(\bar{d}) - 1$ as in Table 1. A negative cell means the arm is ahead in point estimate. $^*$: the one-sided exact sign-flip test rejects after Holm within the column's family at $0.025$. $=$: the arm has the reference's discovery AUC on every seed. Generated by `confirm_latex.py`.

| | SELECTABLE, vs $\text{CoreSet}_{\text{af}}$ | | REMOVED, vs CoreSet | | | |
|---|---|---|---|---|---|---|
| Arm | D1 | D2 | D1 | D2 | D3 | D4 |
| $\text{CoreSet}_{\text{af}}$ | ref | ref | $=$ | $=$ | $=$ | $=$ |
| CoreSet | $=$ | $=$ | ref | ref | ref | ref |
| $\text{CoreSet}_{\text{xf}}$ | +9.5%$^*$ | +32.3%$^*$ | $=$ | $=$ | $=$ | $=$ |
| Random | +304.7%$^*$ | +621.4%$^*$ | +135.2%$^*$ | +106.0%$^*$ | +195.5%$^*$ | −25.7% |
| $\text{Random}_{\text{xf}}$ | +258.9%$^*$ | +1156.1%$^*$ | +142.0%$^*$ | +137.8%$^*$ | +198.0%$^*$ | −36.0% |
| $k$NN distance | +16.7%$^*$ | +11.8%$^*$ | +10.2%$^*$ | +1.2% | −4.0% | +12.2%$^*$ |
| LOF | +23.8%$^*$ | −17.3% | +0.2% | −5.8% | +23.6%$^*$ | −8.3% |
| Isolation Forest | +341.6%$^*$ | +466.8%$^*$ | +161.6%$^*$ | +14.0%$^*$ | +69.8%$^*$ | +25.0%$^*$ |
| NovelMargin | +43.2%$^*$ | +33.0%$^*$ | +5.4%$^*$ | +0.8% | +1.2% | +9.4%$^*$ |
| MixedHalf | +24.6%$^*$ | +30.1%$^*$ | +15.7%$^*$ | +10.4%$^*$ | +13.3%$^*$ | −19.1% |
| MaxHerding | +100.4%$^*$ | +888.6%$^*$ | +38.2%$^*$ | +31.6%$^*$ | +20.1%$^*$ | −8.9% |
| NNDB | +263.6%$^*$ | +4060.5%$^*$ | +131.0%$^*$ | +47.2%$^*$ | +3908.5%$^*$ | −35.9% |
| BADGE | +171.3%$^*$ | +400.6%$^*$ | +29.2%$^*$ | −9.4% | +10.7%$^*$ | −61.5% |
| TypiClust | +1147.9%$^*$ | +2205.2%$^*$ | +338.5%$^*$ | +179.8%$^*$ | +343.4%$^*$ | −6.3% |
| ProbCover | +690.1%$^*$ | +7106.1%$^*$ | +143.8%$^*$ | +143.0%$^*$ | +250.7%$^*$ | −20.0% |
| $k$-means++ | +880.2%$^*$ | +3944.9%$^*$ | +303.0%$^*$ | +323.6%$^*$ | +283.6%$^*$ | −13.3% |
| Holm rejections | 14/15 | 13/15 | 12/15 | 9/15 | 11/15 | 3/15 |

## T EXPLORATORY ANALYSES OF THE CONFIRMATORY RUNS

Everything in this appendix is exploratory and post hoc. `diagnostics.py` and `skip_ladder.py` in `scripts/confirm` ran after the verdicts were known. They read only the committed confirmatory runs and pinned inputs, and write to `data/evidence/exploratory`. A percentile places an item's cosine distance to its 10th nearest neighbor within the distribution of that distance over the labelable items.

**Pool-convention comparison.** SELECTABLE beats NAIVE for 10 of the 12 non-random selectors on D1 and 8 on D2 after Holm at $\alpha = 0.025$ (Table T.1). $k$-means++ and BADGE gain on neither, and random does not gain significantly ($p = 0.30$ and $0.082$). On D1 and D2, the registry's SELECTABLE CoreSet is $\text{CoreSet}_{\text{af}}$. That is, it keeps failed picks as anchors and makes the same picks as $\text{CoreSet}_{\text{af}}$ on every seed. CoreSet's entry in these counts thus includes anchoring. This contrast does not isolate skipping: NAIVE samples its initial pool from labelable rows, whereas SELECTABLE samples from all rows. NAIVE also stops after an entirely failed round and pads its trajectory flat; SELECTABLE continues. The three H2 arms share identical initial pools on every seed. $\text{CoreSet}_{\text{af}}$ versus $\text{CoreSet}_{\text{xf}}$ isolates anchoring, including failures in the initial pool.

**Where rare items sit (H1).** The median percentile of rare-class items is 87.7 on D1, 66.8 on D2 and 93.3 on D3, and the share of them above the labelable rows' 90th percentile is $43.7\%$, $32.4\%$ and $62.2\%$. On D4, a slice of one plant family, the median is 48.1 and the share $6.0\%$. So D4's rare items sit in the bulk of the embedding, and the density family is ahead there. The pattern fits a simple account: distance helps where rare classes are outliers in the embedding, not where they sit among common relatives. The preregistered test does not establish this account, and four datasets cannot. The PlantCLEF rerun (Appendix T) shows that the D4 sign depends on the encoder.

Table T.1: Exploratory and post hoc: the convention ladder. Mean discovery AUC over 16 seeds for CoreSet in three forms: under NAIVE, skipping failed picks ($\text{CoreSet}_{\text{xf}}$), and also anchoring on them ($\text{CoreSet}_{\text{af}}$). Each step shows its gain $\exp(\bar{d}) - 1$, and every step has exact sign-flip $p = 2^{-16}$. Initialization and stopping also change at the first step. Last row: non-random selectors whose AUC rises from NAIVE to SELECTABLE after Holm.

| | D1 iWildCam | D2 Serengeti |
|---|---|---|
| NAIVE CoreSet | 322.6 | 42.5 |
| $\text{CoreSet}_{\text{xf}}$ | 392.2 (+21.5%) | 53.6 (+27.5%) |
| $\text{CoreSet}_{\text{af}}$ | 429.4 (+9.5%) | 71.2 (+32.3%) |
| SELECTABLE beats NAIVE | 10 of 12 | 8 of 12 |

**Isolation on D4 (H1).** Among D4's labelable rows, the 5% with the largest $k$NN distance hold rare-class items at 0.43 times their base rate, against 5.4, 5.3 and 8.2 times on D1 to D3. Under REMOVED, 1.1% of CoreSet's picks on D4 are rare-class against 1.4% of random's. The embedding also separates D4's rare species poorly. Take the rare-class items whose class has at least two items: the nearest neighbor shares the class for 0.14 of them, against 0.39, 0.41 and 0.43 on D1 to D3. On the drone pool, the plant-specific PlantCLEF encoder raises two of three quantities. These values come from the downstream-protocol runs of Appendix Q, not from Table L.1. PlantCLEF raises the nearest-neighbor same-class rate from 0.10 (DINOv2) and 0.13 (DINOv3) to 0.17 and CoreSet's lead over random from 37% and 35% to 43%, but not the rare enrichment among the most isolated labeled items (2.0 times, against 2.5 and 3.1). This agrees with fine-tuned classification on close-ups from the same site, where a plant-specific encoder beats DINOv3 in macro F1 (Saha et al., 2026). The PlantCLEF rerun below suggests that separability alone does not explain the D4 reversal. One account fits the H1 reversal on D4: where isolation does not track class, distance ranks unusual photos of common species above rare ones. A post hoc look at one dataset cannot establish this. The evidence is in `data/evidence/exploratory/nep_separability`.

**D4 with PlantCLEF embeddings (H1).** Post hoc, we rerun D4's H1 arms under REMOVED with PlantCLEF2024 embeddings in place of DINOv3. The manifest, seeds and initial pools are the same. We committed the hypothesis and tooling (`349ab9d`) before the rerun. The rerun ran locally (Python 3.12, scikit-learn 1.9.0), not on the cluster of the confirmatory runs (Python 3.11, scikit-learn 1.9.1). Random does not use the embedding and reproduces its confirmatory mean AUC of 192.2 exactly. The distance-versus-density contrast moves from −15.2% (distance ahead on 0 of 16 seeds) to +7.9% (12 of 16, exact $p = 0.00084$). CoreSet against random moves from −25.7% (0 of 16) to +17.9% (14 of 16). LOF and ProbCover stay behind random (−12.8% and −4.2%, from −19.0% and −7.1%). The rare classes do not become more separable. For rare-class items, the nearest-neighbor same-class rate falls from 0.14 to 0.10, leave-one-out nearest-centroid accuracy from 0.236 to 0.168 and 5-NN accuracy from 0.060 to 0.027. Isolation rises instead. Rare enrichment among the 5% most isolated rows goes from 0.43 to 1.19 times the base rate. On the drone pool, PlantCLEF moved the two the other way: it raised separability and lowered isolation enrichment (above). CoreSet's gain over random rose on both pools. So the D4 reversal depends on the encoder. Neither separability nor isolation alone explains the distance advantage across the two pools. This is one dataset and one encoder swap, examined post hoc. The evidence is in `data/evidence/exploratory/nep_plantclef`.

**The re-proposal trap (H3).** Under NAIVE, 78% of CoreSet's round-10 picks on D1 re-propose a failed pick, which is 94% of its failed picks in that round; on D2 the re-proposal share reaches 95% by round 5, and runs stop after a mean of 4.1 rounds. They stop because a whole round returns no label. Random still finds fewer rare classes within the horizon: at round 10 CoreSet has found 39.9 rare classes on D1 against random's 17.8, and 4.4 against 1.6 on D2. On BCI random overtakes CoreSet at round 4 (26.9 against 26.2). The unlabelable items also sit differently across pools. Their median percentile is 91.5 on D1 but 47.3 on D2 and 66.2 on the drone pool. So unlabelable items lie in sparse regions on D1 only. Even so, CoreSet's failed picks still sit at a median percentile of 99.8 on D2.

Table U.1: **Pool conventions**, named as in the preregistration, with the names they carry in the code and evidence files. Like every convention here, our SELECTABLE runs draw the initial pool from labeled rows with $k = 200$. The preregistered loop draws it from all rows with $k = \max(10, \text{round}(N_{\text{lab}}/40))$. CoreSet$_{\text{xf}}$ only skips failed picks. CoreSet$_{\text{af}}$ also anchors on them, with probe weight $\beta = 0$.

| Convention | Unlabelable items | Failed pick | Code name (evidence) |
|---|---|---|---|
| REMOVED | dropped before the run | cannot occur | protocol `pool_only` |
| COVERED | passed as labeled anchors | cannot occur | protocol `covered` |
| NAIVE | stay candidates | spends budget, re-proposed | protocol `naive_held_out_selectable`; arms `coreset`, `random` in `labelability/` |
| SELECTABLE | stay candidates | spends budget, never a candidate again | arms `coreset_exclude_failed` (CoreSet$_{\text{xf}}$), `labelability_coreset` (CoreSet$_{\text{af}}$), `random_exclude_failed` (Random$_{\text{xf}}$) |

**A learned probe.** An exploratory arm that fits the logistic labelability probe ($\beta = 0.5$, without anchoring) adds little over CoreSet$_{\text{af}}$. It gives a mean log ratio $\bar{d} = +0.056$ on D2 ($p = 0.0022$, uncorrected) and $+0.008$ on D1 ($p = 0.114$). The evidence is in `data/evidence/exploratory/probe_arm`.

**A learned labelability filter.** An exploratory check on the drone pool under SELECTABLE asks whether predicting labelability beats anchoring on failed picks. Each round, a class-balanced logistic regression is fit on the labeled rows against the failed picks. Farthest-first selection then uses the predicted labelability in one of two ways. The filter variant drops candidates with predicted labelability below 0.5, but fills an undersized batch from those candidates. Both variants use skip-only CoreSet until successes and failures have been observed. The weight variant multiplies each distance by the predicted labelability. The harness reproduces the CoreSet$_{\text{xf}}$ and CoreSet$_{\text{af}}$ runs of Appendix C.2 seed for seed. Neither variant significantly beats CoreSet$_{\text{af}}$ on any encoder (exact sign-flip, Holm over 12 one-sided tests). On DINOv3 and DINOv2, both trail CoreSet$_{\text{af}}$ by 9.5 to 12.7% and win on at most 1 of 16 seeds. They trail CoreSet$_{\text{af}}$ even though they cut failed picks by 150 to 202 per run. On PlantCLEF they do not differ significantly from CoreSet$_{\text{af}}$ ($-0.6\%$ and $+1.6\%$) and beat CoreSet$_{\text{xf}}$ by 5.7% and 8.1%. Avoiding predicted failures saves labeling budget but, on two of three encoders, finds fewer rare classes than anchoring on the failures. The design, code and per-seed results are in `data/evidence/exploratory/learned_labelability`.

## U REPRODUCIBILITY ENVELOPE AND REPRODUCTION COMMAND

The PlantCLEF2024 encoder is the checkpoint `vit_base_patch14_reg4_dinov2_lvd142m_pc24_onlyclassifier_then_all`. Table U.1 maps each pool convention to its name in the code and evidence files.

Figure U.1 places the audit in the package's wider command-line workflow.

For a given seed, every selector starts from the same random initial pool. It runs the same $K$ rounds at the same budget and is scored against the same rare-class set. Every run emits a JSON envelope with the library version, the SHA-256 of the embedding file, the seed list, and the per-round trajectories. The driver scripts are `run_evidence_n16.py` (COVERED), `run_pool_only_n16.py` (REMOVED), `run_naive_pool_n16.py` (NAIVE) and `run_baselines_n16.py` (the extended-family baselines). Three commands recompute the statistics from the saved trajectories:

```
python scripts/audit_stats.py --check
python scripts/make_main_tables.py --check
```

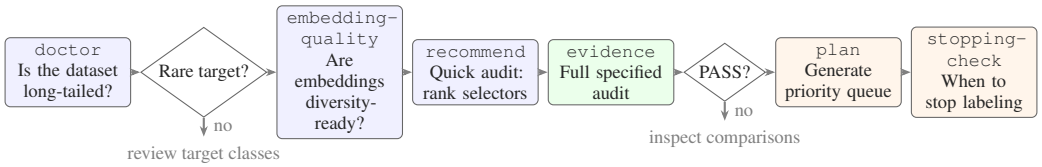



Figure U.1: **Practitioner workflow around the audit.** This paper evaluates only the `evidence` step. The other commands are package conveniences. Each box is a single CLI command, and the main row is the "yes" path. The audit evaluates a specified comparison on labeled retrospective data. A pass does not validate performance in a future campaign.

```
python scripts/check_claims.py --check
```

Each exits nonzero if any number drifts from the committed evidence. The `labelfirst evidence` CLI scores the trapezoidal AUC. The sum over rounds needs the Python API: `audit` with `auc_fn=SeedRun.auc_unique_rare`.

## V Status of every analysis

Table V.1 lists every analysis in the paper and when we fixed it relative to its data. *Recorded before data*: we fixed the analysis before its runs existed. *Post hoc*: we chose or changed it after related results were known. *Exploratory*: we added it to probe or extend a result, and it is not part of the audited claims.

## W Random-number generation at the pinned commit

At library commits `058e4ab` and `575ac75`, four code paths create their generator from the seed inside every selection call. They are random, MixedHalf, ProbCover's tie-break and TypiClust's fallback. So every round draws from the same stream and picks nearly the same relative positions in the remaining pool. These arms are therefore not independent uniform draws across rounds. The numbers in the paper stand as run, since the protocol fixes the code at the pinned commit. H2 compares $\text{CoreSet}_{\text{xf}}$, $\text{CoreSet}_{\text{af}}$ and $\text{Random}_{\text{xf}}$ and uses none of the affected code. We reran each affected arm with the generator seeded from the seed and the size of the labeled set, so each round draws afresh (`data/evidence/rng_sensitivity`). Without the patch, the same harness reproduces the drone-pool audit byte for byte. On the additional datasets it reproduces every pick and every discovery result exactly; the linear-probe accuracies differ from the cluster runs by at most 0.05 (mean) and 0.34 (class-balanced) points in any gain over random. `SOURCE.json` holds the SHA-256 hashes.

On the confirmatory datasets we reran random, MixedHalf and ProbCover, which enter H1 and H3. H1, H2 and H3 keep their verdicts on all three endpoints. The H1 meta-analysis moves from $+122.0\%$ ($p = 0.045$) to $+121.3\%$ ($p = 0.048$). The Removed meta-analysis of H3 moves from $+122.5\%$ ($p = 0.026$) to $+140.8\%$ ($p = 0.012$). But H3 is an intersection, and its Naive component still fails ($p = 0.908$). In the secondary families (Table S.2), CoreSet's lead over MixedHalf on D2 under Removed shrinks from $+10.4\%$ to $+6.7\%$ and is no longer rejected after Holm; no other secondary decision changes. We also dropped the 129 D3 images that overlap the PlantNet subset (disclosure (ix), Appendix X). Without them, local reruns give $+230.8\%$ on D3 as run and $+214.8\%$ with the fresh stream. The meta-analysis then moves from $+120.4\%$ ($p = 0.045$) to $+117.7\%$ ($p = 0.048$); H1 stays not confirmed. The local as-run value differs from the reported +228.3%, because TypiClust and $k$-means++ trajectories differ through KMeans numerics. The other arms we checked reproduce the committed trajectories.

On the drone pool we reran random, MixedHalf and ProbCover under every pool convention and encoder (Table W.1). 0 decisions change under Removed. Under Naive on DINOv3 CoreSet no longer beats random or MixedHalf on the oracle-normalized AUC, so that endpoint's joint claim fails. On DINOv3, in the core family ProbCover leaves CoreSet's Holm rejections on discovery AUC. And on DINOv2 MixedHalf joins the early-AUC Holm rejections. Under Covered on DINOv3,

Table V.1: **Status of every analysis.** We committed the statistics script's constants (margin, endpoints, families, $\alpha$) before its implementation. They apply to every audited row.

| Analysis | Status |
|---|---|
| REMOVED, DINOv3 (Appendix C.3) | Post hoc: made primary after the COVERED results were known |
| REMOVED, DINOv2 and PlantCLEF (Appendix C.3) | Recorded before data |
| COVERED, all encoders (Appendix I) | Post hoc: first primary, then demoted after its results were known |
| NAIVE (Appendix C.2) | Exploratory |
| SELECTABLE, skip-only and anchored (Appendix C.2) | Exploratory |
| Density-term ablations and label-free ProbCover (Appendix C.4) | Post hoc |
| CoreSet against label-free ProbCover (Appendix C.3) | Post hoc, outside the recorded families |
| Pool leave-one-block-out and rare threshold (Appendix J) | Exploratory |
| H1 contrast applied post hoc to the drone pool (Appendix K) | Exploratory, post hoc: preregistered contrast applied to the drone pool |
| Novel-only endpoint, round 1, annotation efficiency (Appendix I) | Exploratory |
| Five-seed replay (Appendix M.1) | Exploratory |
| Embedding geometry and audio (Appendix P) | Exploratory, selected retrospectively |
| Additional datasets, gradient BADGE, downstream accuracy (Appendix Q) | Exploratory |
| CIFAR-100-LT convention check (Appendix Q) | Exploratory, post hoc; design committed before its runs |
| Synthetic analyses and stopping criterion (Appendices F and R) | Exploratory |
| Confirmatory H1 to H3, D1 to D5 (Table 1) | Preregistered, frozen at `367064b`, amended before any confirmatory run |
| Secondary endpoints and families of H1 to H3 (Tables S.1 and S.2) | Preregistered, no confirmatory weight |
| D3 without the 129 exploratory-overlap images (disclosure (ix), Appendix X) | Exploratory sensitivity |
| Diagnostics and skip ladder on the confirmatory runs (Appendix T) | Exploratory, post hoc: after the verdicts |
| CLIP rerun on D1 to D4 (Appendix Q) | Exploratory, post hoc: after the verdicts |
| RNG rerun at the pinned commit (Appendix W) | Exploratory, post hoc |
| D4 rerun with PlantCLEF embeddings (Appendix T) | Exploratory, post hoc: after the verdicts; hypothesis committed before the rerun |
| Tool survey and scikit-activeml reproduction (Appendix O) | Exploratory |
| Mixing-selector sweep (Appendix Q) | Post hoc, exploratory |
| Learned labelability filter on the drone pool (Appendix T) | Exploratory, post hoc |
| Center interpolation and camera-trap neighborhoods (Appendix D) | Exploratory; synthetic design frozen before its runs, neighborhoods descriptive |
| MegaDetector filtering on camera-trap pools (Appendix E) | Exploratory; design frozen before detector inference |

CoreSet and NovelMargin become TOST-equivalent on the trapezoid AUC. With the fresh stream, ProbCover gains up to $4.0\%$ over random under REMOVED, against $1.7\%$ as run. On DINOv2 its one-sided exact sign-flip $p$ against random falls from $0.160$ to $0.021$, below $\alpha = 0.025$ unadjusted but not after Holm over the three encoders ($0.064$). In local reruns of all $160$ confirmatory TypiClust runs, TypiClust's fallback never fired. These reruns differ from the committed trajectories through KMeans numerics.

On the additional datasets (Appendix Q), we reran random, MixedHalf and ProbCover on iNaturalist, PlantNet and the three drone-pool encoders. TypiClust keeps its committed runs, since the patch leaves its local discovery picks unchanged. CoreSet's gain over random in discovery AUC moves from $+64.26\%$ to $+59.21\%$ on PlantNet and by at most $1.29$ points on the other rows. The core family's joint claim still passes on PlantNet and fails on iNaturalist. CoreSet's mean test accuracy

against random moves from $-0.96\%$, $-1.41\%$ and $+0.38\%$ to $-1.75\%$, $-1.87\%$ and $+0.25\%$ on the drone-pool encoders, from $+1.18\%$ to $+0.31\%$ on iNaturalist and from $-9.47\%$ to $-9.67\%$ on PlantNet. Its class-balanced accuracy against random on the drone-pool encoders spans $-2.13\%$ to $+2.26\%$, against $-0.41\%$ to $+3.30\%$ as run (Section 6). NovelMargin and gradient BADGE still lead random on class-balanced accuracy on all five rows, by $+5.83\%$ to $+11.37\%$. We did not re-run the mechanism, sensitivity, labelability, mixed-ratio, mixing-selector, separability and synthetic analyses, CIFAR-100-LT, the CLIP rerun or the D4 PlantCLEF rerun. Their random-based numbers carry the same caveat. The library's development branch draws a fresh stream each round.

Table W.1: **Fresh random stream per round**, drone pool, discovery AUC, $n_s = 16$. Gain over random in percent, as run at the pinned commit and with a fresh stream each round. CoreSet rows give CoreSet's gain. MixedHalf and ProbCover rows give that selector's own gain.

| | DINOv3 | | DINOv2 | | PlantCLEF | |
|---|---|---|---|---|---|---|
| | as run | fresh | as run | fresh | as run | fresh |
| CoreSet, NAIVE | −27.23 | −32.15 | −32.51 | −37.08 | −49.51 | −52.93 |
| CoreSet, COVERED | +39.11 | +38.69 | +40.54 | +40.12 | +45.61 | +45.17 |
| CoreSet, REMOVED | +46.26 | +47.48 | +46.81 | +48.03 | +50.60 | +51.85 |
| MixedHalf, REMOVED | +30.52 | +33.90 | +31.00 | +33.74 | +32.59 | +34.75 |
| ProbCover, REMOVED | +0.60 | +2.98 | +1.67 | +3.99 | +1.60 | +1.83 |

## X ANALYSIS HISTORY AND DISCLOSURES

**Ineligible dataset.** The protocol names a fifth dataset, RFCx audio (D5). It is ineligible (332 labelable windows of 1,000 required, 2 rare classes of 10); we report it and do not replace it. The protocol left its manifest hash pending and no amendment froze it; the eligibility check hashed a manifest that is not in the archive.

**Primary estimand on the drone pool.** The study first made COVERED primary. Under COVERED, CoreSet did not beat NovelMargin, and the joint claim failed on every encoder (Table M.2). We then switched the primary estimand from COVERED to REMOVED, after those results were known. We also knew the REMOVED result on DINOv3 when we switched. We committed the statistics script's constants (margin, endpoints, families, $\alpha$) before its implementation. The DINOv2 and PlantCLEF REMOVED runs are the only drone-pool analyses fixed before their data existed. They share their images and labels with DINOv3.

**Names in the code and evidence files.** The evidence files store the core family as `submitted` and the extended family as `revised`. They keep the legacy key `badge` for the $k$-means++ selector. `mechanism_check.py` checks the density-term ablations. For random, CoreSet and NovelMargin, the SELECTABLE runs of Appendix C.2 reproduce the NAIVE trajectories exactly. `check_claims.py` verifies this before using them.

**Disclosures.** (i) Two endpoints fail under REMOVED: final $u_K$ on PlantCLEF (4/6) and early AUC on DINOv2 (5/6), Appendix L. (ii) Under COVERED, CoreSet is 3.1 to 8.5% behind NovelMargin in early AUC (rounds 1 to 3). (iii) $k$NN distance ties CoreSet on DINOv2 ($p = 0.060$) and is close elsewhere. (iv) The reverse-direction NovelMargin result under COVERED is exploratory. NovelMargin's advantage has reverse $p = 0.020$ on DINOv2 and 0.037 on PlantCLEF, against forward $p = 0.98$ and 0.97. (v) A rerun at the pinned commit replaced the PlantNet gradient-BADGE trajectories in the audit statistics but not in Table Q.1 (Appendix Q). (vi) ProbCover's purity radius uses the ground-truth class count. The main ProbCover comparison uses a label-free radius (Appendix C.3). (vii) We made REMOVED primary post hoc on the drone pool (Appendix C). The protocol fixed REMOVED for H1 before any confirmatory run. (viii) We froze the protocol at `367064b` and amended it at `b4f2c88`, `4ff2867`, `6c2ba3c`, `fe3dc97`, `c5bf28b` and `725da03`. All amendments came before the runner `f9ce6e2` and any confirmatory run. Amendment `6c2ba3c` moved H3's second arm from SELECTABLE to NAIVE, since the first wording contradicted H2b. (ix) 129 D3 images also appear in the exploratory PlantNet subset (Appendix Q). Without them, in an exploratory check, H1 stays not confirmed ($p = 0.045$). (x) The preregistered

secondary families do not all favor the reference (Appendix S). CoreSet beats only 3 of 15 arms on D4, and LOF is ahead of $\text{CoreSet}_{\text{af}}$ in point estimate on D2.