MAKING GENERALIZED CATEGORY DISCOVERY USEFUL FOR BIODIVERSITY MONITORING

Anonymous authors
Paper under double-blind review

ABSTRACT

By organizing previously unseen categories into coherent groups, Generalized Category Discovery (GCD) can potentially facilitate biodiversity monitoring by organizing observations of new species, enabling models to work with data-deficient species, and accelerating individual re-identification. Yet GCD methods are rarely designed to bear in mind the goals and constraints of biologists or the challenges posed by biodiversity data. This raises a practical question: how can we make GCD useful for biodiversity monitoring? To address this question, we formulate three biodiversity monitoring tasks that can benefit from GCD and propose practical metrics to evaluate their downstream performance under budget constraints. We introduce `Sciurus`[1], a GCD-based sample selection framework for identifying informative samples, and show that it substantially improves downstream performance over existing sample selection methods. We further find that clustering accuracy, the most commonly used evaluation metric for GCD, is not always a reliable proxy for downstream utility, suggesting that selecting models based solely on clustering accuracy may be suboptimal for downstream applications. To address this gap, we introduce *Homogeneity*, a complementary metric that measures within-cluster consistency to better predict downstream performance. Together, our work provides a concrete first step toward making GCD useful for biodiversity monitoring in practice.

1 INTRODUCTION

Biodiversity is declining at an unprecedented rate (Cowie et al., 2022), yet our understanding of Earth's species remains extremely limited. Automated monitoring systems and community science platforms are generating vast amounts of biodiversity data, creating unprecedented opportunities to discover new species and monitor their population change at scale. However, the volume of these data far exceeds the capacity of domain experts to analyze them manually (Mulero-Pázmány et al., 2025; Thornton et al., 2026). Machine learning (ML) has been increasingly used to accelerate the process (Tuia et al., 2022). In practice, ML models will inevitably encounter categories unseen during training for several reasons. First, an estimated 86% of terrestrial and 91% of marine species remain undocumented (Mora et al., 2011; Barnosky et al., 2011). Second, many described species lack sufficient data for ML model training. For example, BioCLIP 2, a large-scale foundation model (Gu et al., 2025) was trained on over 214M images spanning 952K species, yet this represents less than half of the 2.3M known species Bánki et al. (2026); and even among included species, the data is strongly long-tailed. In the related task of individual re-identification (reID), new individuals are continually born and enter monitored populations (Adam et al., 2025). Hence, the need of recognizing and organizing new categories is not an edge case but an inherent requirement for biodiversity monitoring.

Generalized Category Discovery (GCD) offers a promising approach to address this challenge. Unlike clustering, which only has access to unlabeled data, GCD learns from both labeled and unlabeled data to organize unlabeled observations into coherent clusters, including previously unseen categories (Vaze et al., 2022). In an ideal scenario, the clusters produced by a GCD model would perfectly align with the underlying categories of interest, such as previously unknown species or individuals.

[1] *Sciurus* is the genus name of fox squirrels (*Sciurus niger*), which are known to organize their food caches by nut type (Delgado & Jacobs, 2017).

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.

Labeling a single representative from each cluster could then enable the discovery of all new categories and the annotation of all observations at a fraction of the cost of labeling images individually. In practice, however, state-of-the-art GCD models remain far from perfect, especially when applied to fine-grained, long-tailed biodiversity data (He et al., 2025), such as images of specimens from natural history collections (Tan et al., 2019). The sheer volume of biodiversity data also imposes practical constraints on human verification: experts often cannot verify the entire dataset and must instead select a limited subset of informative observations to answer ecological questions given their budget. Consequently, sample selection is an integral part of the machine-assisted monitoring pipeline. Yet, it remains unclear to what extent clustering performance affects performance on downstream biodiversity monitoring tasks. For example, as illustrated in Figure 1 (Bottom Left), discovering all new categories does not require every observation to be assigned to the correct cluster. It may be sufficient to select one representative observation from each category for annotation. Thus, downstream discovery performance depends not only on clustering quality, but also on how effectively clusters facilitate sample selection for human reviews.

To evaluate the potential of current GCD methods for biodiversity monitoring in practice, we formulate three applications that could benefit from GCD: (1) discovering new categories, (2) efficiently annotating samples from new categories, and (3) training classifiers to recognize these new categories, illustrating them in Figure 1. We introduce practical evaluation metrics that reflect biologists' need in practice and propose `Sciurus`, a GCD-based sample selection framework for identifying informative samples under limited annotation budgets. Across these applications, we find that `Sciurus` consistently outperforms existing sample selection strategies, demonstrating that GCD can be useful for biodiversity monitoring even when its clusters are imperfect.

However, we also find that the effectiveness of `Sciurus` depends strongly on the underlying GCD model. We investigate this deeper and find that clustering accuracy, the mostly commonly used metric for clustering quality, is not always a reliable proxy for downstream performance. To address this, we introduce *Homogeneity*, which measures the within-cluster consistency of observations. Across biodiversity monitoring tasks, *Homogeneity* is a stronger proxy for downstream performance than clustering accuracy, offering a complementary way to assess GCD models. Overall, our formulation of the three practical tasks, the introduction of `Sciurus`, and the analysis on clustering quality provide a concrete first step toward developing GCD methods for biodiversity monitoring in practice.

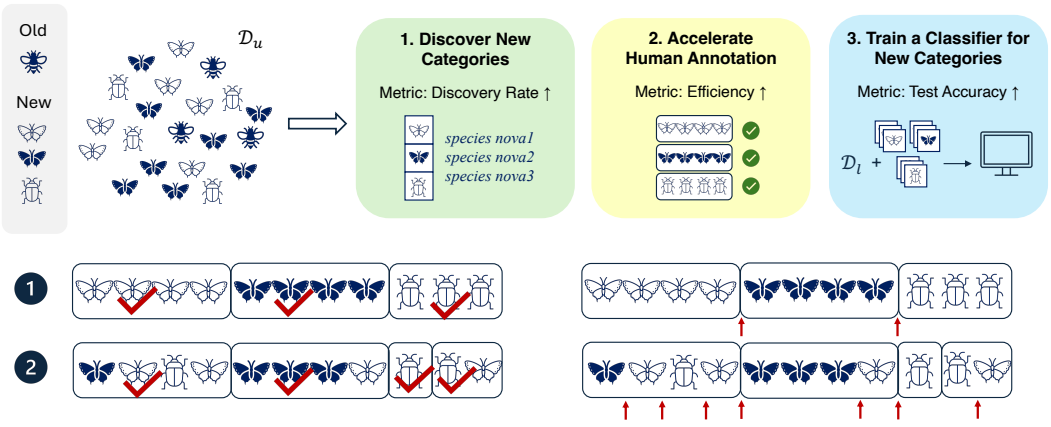

Figure 1: **Top: Three Applications of Generalized Category Discovery (GCD) for Biodiversity Monitoring.** We consider three common biodiversity monitoring tasks that GCD can facilitate and define a practical evaluation metric for each application. **Bottom Left: Clustering Quality vs. Discovery Rate.** Black boxes indicate clusters and red ticks indicate selected samples. Cases 1 and 2 represent perfect and imperfect clustering, respectively. Selecting samples closest to cluster centers achieves perfect discovery in both cases, despite the difference in clustering quality. **Bottom Right: Two Verification Streams.** Black boxes indicate clusters and red arrows indicate where a switch occurs. Stream 1 achieves the minimum switches, while stream 2 has more switches.

2 BACKGROUND

2.1 ML-ASSISTED DATA PROCESSING WORKFLOW IN BIODIVERSITY MONITORING

Several ML models have been developed image-based biodiversity monitoring, such as animal detection (Beery et al., 2019; Svenning et al., 2026), species recognition (Stevens et al., 2024; Gu et al., 2025), and individual reID (Berger-Wolf et al., 2017; Adam et al., 2025). In practice, biologists have started to use ML to accelerate data annotation. For example, Bigham & Carter (2026) use model predictions to drastically decrease the amount of time needed for data annotation (from an estimated 5.9 years to 6 weeks) and iteratively improve the model performance with newly annotated data. Szczygieł et al. (2026) demonstrate a practical workflow which clusters images based on visual similarity. Under this setting, better clustering quality can lead to better organization of data, and hence improves verification efficiency. Inspired by these workflows, we outline three applications that can benefit from GCD. All applications are subject to an annotation budget b, requiring a subset A with $|A| = b$ to be selected from a large unlabeled pool.

Application 1 (Discover New Categories). This is a straightforward application of GCD: to help biologists efficiently discover new categories from a large pool of data. Oftentimes, the budget is so limited that it only allows biologists to verify at most one image per category. Hence, for this application, we consider a very low-budget setting where b is no greater than $|\mathcal{Y}_{\text{new}}|$. A GCD method is considered better if it achieves a higher discovery rate, defined as

$$\text{DiscoveryRate(A)} = d/|\mathcal{Y}_{\text{new}}|,$$

where d is the the number of new categories in the selected samples.

Application 2 (Accelerate Human Annotation). When annotating a subset of data, we often focus primarily on the outcome of the annotation process, but overlook the annotation process. In practice, the same set of samples can lead to substantially different annotation cost depending on how the data are presented to the annotator. Thus, it is also important to measure the annotation efficiency. Grouping related images together is a common practice in computer vision to accelerate annotation (Moore & Corso, 2020). This approach is adopted by biologists as well (Szczygieł et al., 2026). Cluster outputs of GCD provide a natural grouping of images. Hence, we evaluate its performance based on the annotation efficiency.

Since reviewing images belonging to the same category is usually much faster than randomly ordered images due to fewer context switches, annotation time should be proportional to the number of context switches. We therefore measure the annotation time in terms of the number of context switches, which is defined as as

$$S(A) = \sum_{i=2}^{n} \mathbb{I}(y_i \neq y_{i-1}),$$

for a set A and y_i denotes the ground-truth category of the i-th image in A (see Figure 1 for an illustration). Let c denote the number of unique categories in A. The minimum number of switches is $S_{\min} = c - 1$, achieved when images are grouped consecutively by category. We define annotation efficiency as

$$\text{Efficiency}(A) = (|\mathbf{y}| - S(A))/(|\mathbf{y}| - S_{\min}),$$

where $\mathbf{y}$ denotes the sequence of ground-truth labels in A. This efficiency is 1 when all images from each category are grouped consecutively and decreases as the number of category switches increases.

Using *Efficiency* alone, however, can be misleading. For example, if A contains samples from only a single category, *Efficiency* is 1 by definition, despite discovering only one new category. We therefore study the trade-off between discovery rate and efficiency using Pareto dominance: for Application 2, a method is Pareto-optimal if no other method achieves both higher discovery rate and higher efficiency. To summarize the two objectives with a single metric, we define *Effectiveness* as their average:

$$\text{Effectiveness} = (\text{Discovery Rate} + \text{Efficiency})/2.$$

Application 3 (Train a Classifier for New Categories). Finally, expert-verified images provide training data for classifiers of newly discovered categories. We evaluate this application by measuring the resulting classifier's accuracy on a held-out test set, denoted *Test Accuracy*.

2.2 Generalized Category Discovery

The term *Generalized Category Discovery* was coined by Vaze et al. (2022) as an extension to Novel Category Discovery (Han et al., 2019). Consider a dataset $\mathcal{D}$ consisting of a labeled subset $\mathcal{D}_l$ and an unlabeled subset $\mathcal{D}_u$, where

$$\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \in \mathcal{X} \times \mathcal{Y}_l, \tag{1}$$

$$\mathcal{D}_u = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M} \in \mathcal{X} \times \mathcal{Y}_u, \tag{2}$$

and $\mathcal{Y}_u \setminus \mathcal{Y}_l \neq \emptyset$. During training, the model is trained with images $\mathbf{x}_i$ and labels y_i from $\mathcal{D}_l$ and images $\mathbf{x}_j$ from $\mathcal{D}_u$. During inference, the model predicts the labels $\hat{y}_i$ for images in $\mathcal{D}_u$.

Evaluation. Model performance is evaluated by the clustering accuracy (ACC)

$$\text{ACC} = \max_{p \in \mathcal{P}(\mathcal{Y}_u)} \frac{1}{|\mathcal{D}_u|} \sum_{i=1}^{M} \mathbb{1}\{y_i = p(\hat{y}_i)\}, \tag{3}$$

where $P(\mathcal{Y}_u)$ is the set of all permutations of the labels in the unlabeled set. The optimal assignment between cluster indices and ground truth labels is solved by the Hungarian algorithm (Kuhn, 1955).

3 Related Work

Generalized Category Discovery (GCD). GCD methods can be broadly categorized as parametric or non-parametric. Parametric methods use a classification head for clustering and jointly optimize representation and classification objectives, while non-parametric methods learn representations alone and requires a separate clustering algorithm at inference time. Vaze et al. (2022) propose a non-parametric approach using contrastive learning (Chen et al., 2020; Khosla et al., 2020). Building on this representation-learning framework, SimGCD (Wen et al., 2023) introduces a classification head trained through prototype-based self-distillation, while SEAL (He et al., 2025) extends it with multiple heads for different semantic hierarchies and enforces consistency across hierarchy levels. In contrast, Self-Expertise (Rastegar et al., 2024), a non-parametric method, improves representation learning through hierarchical pseudo-labeling. Other approaches modify the representation space, such as using hyperbolic geometry (Liu et al., 2025), or improve the classification head using equiangular tight frame prototypes (Han et al., 2025).

Sample Selection. Ma et al. (2024) extend GCD to the active learning setting (AGCD) and propose `Adaptive-Novel`, which iteratively selects informative samples to improve clustering performance. Our work differs from AGCD in two key aspects: the objective and the selection method. First, AGCD aims to improve the clustering accuracy, whereas we consider GCD as a means to accomplish downstream tasks rather than an end in itself. Second, `Adaptive-Novel` cannot be applied when the annotation budget is smaller than the number of new categories, limiting its usefulness under tight budgets. With a sufficiently large budget, `Adaptive-Novel` can be viewed as a special case of our proposed selection pipeline `Sciurus`, when both query and sorting strategies are random. More broadly, sample selection has been extensively studied in active learning for classification, leveraging classifier outputs such as features Sener & Savarese (2018), logits Wang & Shang (2014), and gradients Ash et al. (2020) for the query strategy.

4 Datasets

We use three biodiversity-related datasets: CUB (Wah et al., 2011), Herbarium 19 (Tan et al., 2019), and Open Insect (Chen et al., 2025) for our experiments to cover a broad range of taxa (birds, insects, and plants) and diverse image sources (live or museum specimens). Each dataset has four data splits: the labeled dataset $\mathcal{D}_l$, the unlabeled dataset $\mathcal{D}_u$, the validation set $\mathcal{D}_{\text{val}}$, and the test set $\mathcal{D}_{\text{test}}$. Categories in $\mathcal{D}_l$ are considered "Old", while categories that only occur in $\mathcal{D}_u$ (i.e., in $\mathcal{Y}_u \setminus \mathcal{Y}_l$) are considered "New". For CUB and Herbarium 19, we follow the data splits of $\mathcal{D}_l$ and $\mathcal{D}_u$ used in Vaze et al. (2022) and randomly sample 5% of the test images belonging to the "Old" categories to obtain $\mathcal{D}_{\text{val}}$. For Open Insect, we only use a subset of Open-Insect C-America to make sure that the "New" categories do not overlap with species in BioCLIP's training set. We summarize the data distribution in Table 1. Note that in GCD literature, there are only two data splits: $\mathcal{D}_l$ and $\mathcal{D}_u$. Models are trained

on $\mathcal{D}_l$ and $\mathcal{D}_u$ and evaluated on $\mathcal{D}_u$, not $\mathcal{D}_{\text{test}}$. Likewise, our metrics for Applications 1 and 2 are computed on $\mathcal{D}_u$ since our goal is to effectively study a large pool of unlabeled data. $\mathcal{D}_{\text{val}}$ is only used for the filter function f_{filter} in Algorithm 1, whil $\mathcal{D}_{\text{test}}$ is only used for Application 3.

Table 1: **Data distribution.** For each dataset, we report the number of categories with the corresponding number of images shown in parentheses. Categories in $\mathcal{D}_l$ are considered "Old", while categories that only occur in $\mathcal{D}_u$ (i.e., in $\mathcal{Y}_u \setminus \mathcal{Y}_l$) are considered "New". Evaluation is done on $\mathcal{D}_u$ for Applications 1 and 2 and on $\mathcal{D}_{\text{test}}$ for Application 3.

Split	CUB		Herbarium 19		Open Insect	
	Old	New	Old	New	Old	New
Labeled $\mathcal{D}_l$	100 (1.5k)	-	341 (8.9k)	-	591 (24.1k)	-
Unlabeled $\mathcal{D}_u$	100 (1.5k)	100 (3k)	341 (8.9)	342 (16.5k)	590 (24.2k)	1180 (29.8k)
Validation $\mathcal{D}_{\text{val}}$	76 (144)	-	57 (68)	-	216 (689)	-
Test $\mathcal{D}_{\text{test}}$	100 (2.7k)	100 (2.9k)	341 (1.3k)	342 (1.3k)	557 (13.1k)	1151 (12.0k)

5 METHOD

5.1 GENERALIZED CATEGORY DISCOVERY

Estimate the number of categories. Most GCD methods require the number of new categories for training and inference. To reflect a realistic setting, unlike most prior GCD work, we do not assume access to the ground truth of $|\mathcal{Y}_l \cup \mathcal{Y}_u|$ or $|\mathcal{Y}_u|$, and use the agglomerative clustering to estimate these numbers instead. We used the scikit-learn implementation with the Ward linkage and the Euclidean distance metric (Pedregosa et al., 2011). The number of clusters is determined by the distance threshold τ. To calibrate τ, we ran the algorithm on $\mathcal{D}_l$ over a range of threshold values and select the one that produces a number of clusters closest to $|\mathcal{Y}_l|$ using binary search. We experimented with features extracted by three off-the-shelf foundation model, all of which have similar estimates for $|\mathcal{Y}_l \cup \mathcal{Y}_u|$ and $|\mathcal{Y}_u|$ (see Table 4 in the Appendix). For simplicity, for all subsequent experiments, we use the estimates given by DINOv3.

Backbones. We consider two self-supervised foundation models: DINOv2 (ViT-B/14 with registers) (Darcet et al., 2024) and DINOv3 (ViT-B/16 pretrained on LVD-1689M) (Siméoni et al., 2025). We additionally include BioCLIP (Stevens et al., 2024), a supervised model that has been increasingly adopted for biodiversity monitoring. To ensure consistency with the GCD literature, we retain the 'new" categories in CUB and Herbarium 19, including those that overlap with BioCLIP's training set. While such overlap could potentially lead to over-promising results, we find in our experiments that BioCLIP performs comparably to the other backbones on these datasets (see Figure 7 in the Appendix). For Open-Insect, we ensure that none of the 'new" categories overlap with BioCLIP's training set to reflect the reality as labeled data for newly discovered species are unavailable.

Methods. We study three GCD methods that require training: 1) *HypSelEx* (Rastegar et al., 2024; Liu et al., 2025), 2) *SimGCD* (Wen et al., 2023), and 3) *SEAL* (He et al., 2025), representing both parametric and non-parametric approaches. All methods require an estimate of the total number of categories, $|\mathcal{Y}_l \cup \mathcal{Y}_u|$, during both training and inference. In addition, SEAL requires estimates of the number of categories at each level of the hierarchy (shown in Table 5 in the Appendix). We also include two simple baselines that do not require training: 1) *Agglomerative Clustering* and 2) *K-Means*, using features extracted the three backbones off the shelf.

Table 2: **Comparison of methods and their requirements.** The "Training" columns indicates whether the method requires model training. For the "Inference" column, "CLS head" indicates using the classification head for clustering, while other values indicate using the corresponding clustering method for clustering.

Method	Training	Inference
Agglomerative	✗	agglomerative
K-Means	✗	kmeans
SimGCD	✓	CLS head
SimGCD (kmeans)	✓	kmeans
SEAL	✓	CLS head
SEAL (kmeans)	✓	kmeans
HypSelEx (kmeans)	✓	kmeans

Training. For HypSelEx, SimGCD, and SEAL, we trained six model configurations by combining three backbones (BioCLIP, DINOv2, and DINOv3) with two fine-tuning strategies commonly used in GCD

literature: fine-tuning either the last block or the last two blocks. Each configuration is trained with three random seeds, and we report the average performance across runs. All models are trained for 100 epochs, with the final checkpoint used for inference. Additional training details and hyperparameters are provided in the Appendix.

Inference. For all five methods, we use the model as a feature extractor. For HypSelEx, SimGCD, SEAL, and K-Means, we then run k-means on these features with k equal to the estimated $|\mathcal{Y}_u|$. For Agglomerative Clustering, for each backbone, we use the same configuration aforementioned with corresponding τ shown in Table 4. For the two parametric methods, SimGCD and SEAL, we additionally use their classification head for clustering. We summarize each method's training requirement and the inference method in Table 2.

5.2 Sample Selection Pipeline

In Algorithm 1, we propose `Sciurus`, a GCD-based selection pipeline, for informative sample selection. Given $\mathcal{D}_l$ and $\mathcal{D}_u$, we train a GCD model f_{gcd} using $\mathcal{D}_l \cup \mathcal{D}_u$. Unlabeled samples that are likely to belong to old categories are removed using a filter function f_{filter}. The remaining samples are then clustered using a GCD model f_{gcd}. In each iteration, the query strategy μ selects one candidate from each cluster. If we have more selected candidates than the remaining budget b', we sort them with the sorting strategy σ and only keep the first b' candidates. Otherwise, we keep all candidates. The procedure is repeated until b is exhausted.

Algorithm 1 `Sciurus`: Informative Sample Selection with Generalized Category Discovery (GCD)

Require: A labeled dataset $\mathcal{D}_l$, an unlabeled dataset $\mathcal{D}_u$, a filter function f_{filter}, a query function μ, a sorting function σ, an annotation budget b
Ensure: An annotated set A

Train a GCD model f_{gcd} on $\mathcal{D} = \mathcal{D}_u \cup \mathcal{D}_l$
$\mathbb{C}_u \leftarrow f_{\text{gcd}}(\mathcal{D}_u)$ # $\mathbb{C}_u = \{C_1, C_2, \cdots\}$ and $C_i = \{(x_1, i), \cdots, (x_j, i), \cdots\}$
$\mathcal{D}_{u'} \leftarrow f_{\text{filter}}(\mathcal{D}_u)$
Update $\mathbb{C}_u$ to $\mathbb{C}_{u'}$ such that $C_i = \{(x_1, i), \cdots, (x_j, i), \cdots | x_i \in \mathcal{D}_{u'}\}$
$A = \{\}$
while $b > 0$ and $\exists C_i \in \mathbb{C}_{u'}$ such that $|C_i| > 0$ **do**
    $Q \leftarrow \sigma(\mu(\mathcal{C}_{u'}))$
    for $q_i \in Q = \{q_1, \cdots, q_j\}$ **do**
        $A \leftarrow A \cup \{q_i\}$
        $b \leftarrow b - 1$
        if $b \leq 0$ **then**
            break
        end if
    end for
    for $C_i \in \mathbb{C}_{u'}$ **do** # Remove the annotated samples from $C_{u'}$
        $C_i \leftarrow C_i \setminus (A \cap C_i)$
    end for
end while
if $b > 0$ **then**
    Repeat the while-loop for $\mathcal{D}_u \setminus \mathcal{D}_{u'}$
end if
return A

The filter function function f_{filter}, the query strategy μ, and the sorting strategy σ can all make use of scores commonly used in Open-Set Recognition (OSR), which aims to distinguish samples from seen and unseen categories to prioritize "New" categories. To obtain such scores, we train a classifier f_{cls} on $\mathcal{D}_l$ by fine-tuning the last two blocks of DINOv3 with a linear classification head.

The filter function. We use confidence – the maximum softmax probability as the OSR score as it has been shown to perform well on fine-grained biodiversity datasets (Chen et al., 2025). We use $\mathcal{D}_{\text{val}}$ to determine the OSR threshold. Specifically, we set threshold to the 30th percentile of the confidence among samples with a correct top-1 prediction. Unlabeled samples with a confidence higher than the threshold are considered "Old" and removed from $\mathcal{D}_u$, yielding $\mathcal{D}_{u'}$ in Algorithm 1.

Query strategies. We consider four query strategies: 1) Center, 2) Confidence, 3) Margin, and 4) Random. For Center, we select the sample that is closet to the cluster center measured by Euclidean distance. For Confidence and Margin, we select the sample with the lowest confidence or margin (the difference between the largest and the second largest softmax probability) given by f_{cls}. For Random, we select a random sample.

Sorting strategies. We consider five sorting strategies 1) Cluster confidence, 2) Sample confidence, 3) Cluster margin, 4) Sample margin, and 5) Cluster variance. "Cluster" means that we sort the clusters first by the mean confidence or margin of each cluster and select the top n samples according to the cluster ranking. "Sample" means we directly sort the sample confidence or margin and select the top n samples. For cluster $\mathcal{C}_i$, the cluster variance is defined as $\sum_j (\mathbf{z}_j - \bar{\mathbf{z}}_i)^2 / (|C_i| - 1)$, where $\mathbf{z}_j$ is the feature vector extracted by the GCD model and $\bar{\mathbf{z}}_i = \sum_j \mathbf{z}_j / |C_i|$. All sorting strategies sort the values in descending order.

6 RESULTS

Baselines. We compare `Sciurus` to five sample selection methods commonly used in active learning (denoted *Baseline*), namely Margin (Wang & Shang, 2014), Confidence (Wang & Shang, 2014), Entropy (Wang & Shang, 2014), BADGE (Ash et al., 2020), and Coreset (Sener & Savarese, 2018). These methods require a classifier trained on $\mathcal{D}_l$. Specifically, we first train a classifier g on $\mathcal{D}_l$ using each of the three backbones (BioCLIP, DINOv2, and DINOv3) following the setup in Section 5.2. We also consider the simplest method – simple random sampling (denoted *Random*).

The performance of `Sciurus` depends on five components: (1) the GCD method, (2) the backbone, (3) the fine-tuning strategy, (4) the query strategy μ, and (5) the sorting strategy σ. We refer to the Cartesian product of these components as a "configuration." For each application and budget, we report the best performance achieved by `Sciurus` across all configurations and, for comparison, the best performance achieved by any of the five baseline methods. For Application 2, all GCD methods first sort samples by cluster assignment and then by classifier confidence within each cluster using f_{cls}. *Baseline* and *Random* methods use the corresponding classifier g for the two-stage sorting. For Application 3, we apply a 1-nearest-neighbor classifier with Euclidean distance $\mathcal{D}_{\text{test}}$. Results with alternative within-cluster sorting strategies and classification methods are provided in Tables 6 and 7 in the Appendix.

In real-world applications, the budget is determined by practical constraints, such as the amount of time and resources available for annotation. We therefore consider several budget scenarios for each application to assess whether our conclusions remain consistent across budgets. At 100% $|\mathcal{D}_u|$, sample selection is unnecessary, but GCD may still improve efficiency through better clustering. As shown in Figure 2, `Sciurus` achieves the best performance in all scenarios. For Application 2, we also compare methods based on their Pareto dominance (see Figure 4 in the Appendix). A method is Pareto-optimal if no other method achieves both higher discovery rate and higher efficiency. For all budgets and datasets, `Sciurus` is Pareto-optimal compared to *Baseline* and *Random*.

7 DISCUSSION

7.1 CLUSTERING QUALITY AND METRIC CORRELATION

While `Sciurus` consistently outperforms the alternative methods, its usefulness in practice relies on its robustness across configurations. We first analyze the three components that determine the clustering output: the GCD method, backbone, and fine-tuning strategy. We observe that there is a substantial performance gap between different settings (see Figures 5 and 6 in the Appendix), highlighting the importance of model selection. Clustering accuracy (Equation (3)) is the most common evaluation in GCD literature, where it is computed over all categories (ACC All), as well as separately for old and new categories (ACC Old and ACC New). However, we find that these metrics do not consistently predict downstream performance (Figure 7 in the Appendix). Thus, selecting GCD models based solely on ACC may overlook configurations that are more effective for practical biodiversity monitoring.

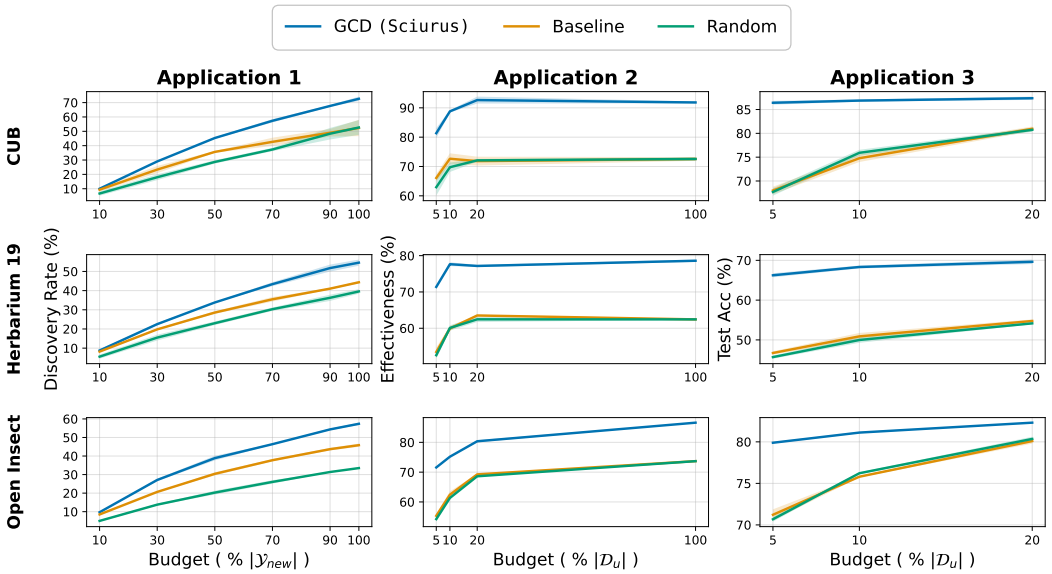

Figure 2: **Performance on the three biodiversity monitoring tasks.** For each application and budget, we consider several budgets and report the best performance achieved by `Sciurus` and the baselines. We also include Random sampling for reference. All results are averaged across three runs and shaded areas indicate $\pm$ one standard deviation across seeds. `Sciurus` outperforms the other methods by a large margin across datasets, applications, and budgets.

One possible explanation is that ACC has two limitations in our setting. First, ACC requires a one-to-one matching between predicted clusters and ground-truth classes, and this mapping can be dominated by categories with more samples. Second, ACC measures whether clusters can be globally aligned with ground-truth classes, whereas our downstream tasks often depend more directly on whether samples assigned to the same cluster belong to the same class. We therefore introduce *Homogeneity*, which measures the proportion of samples in each predicted cluster that belong to its majority ground-truth category. Let $\mathcal{C} = \{c_1, \ldots, c_k\}$ and $\mathcal{Y} = \{y_1, \ldots, y_m\}$ denote the predicted clusters and ground-truth categories, respectively. Let N_{ij} be the number of samples assigned to cluster c_i and class y_j. We define

$$\text{Homogeneity}(\mathcal{C}, \mathcal{Y}) = \frac{1}{N} \sum_{i=1}^{k} \max_{j \in 1,\ldots,m} N_{ij}, \tag{4}$$

where $N = \sum_{i,j} N_{ij}$ is the total number of samples. Unlike ACC, Homogeneity does not require a one-to-one correspondence between clusters and categories, allowing multiple clusters to represent the same ground-truth category. We find that *Homogeneity* is more consistently correlated with downstream performance than ACC All, ACC Old, and ACC New, as measured by Kendall's rank correlation τ between each clustering quality metric and each downstream evaluation metric (Figure 3). This difference is much more noticeable on long-tailed datasets (Herbarium 19 and Open Insect) than on balanced datasets (CUB).

7.2 Beyond Clustering Quality

Query and sorting strategies matter. We then fix the GCD method, backbone, and fine-tuning strategy, and vary either one of the two strategies or their combination to measure the performance gap between the best- and worst-configuration. We find that both μ and σ can substantially affect downstream performance. For example, in Application 1 with a budget of $|\mathcal{Y}_{\text{new}}|$, the average discovery rate gap across datasets is 8% for μ, 14% for σ, and 19% for their combination. At a budget of $0.05|\mathcal{D}_u|$, the average gaps in effectiveness and test accuracy are both 9% for μ, compared with 5% and 2% for σ, respectively, and 10% for both metrics for their combination. These results show that the relative importance of μ and σ depends on the downstream application, and that both are important components of `Sciurus`.

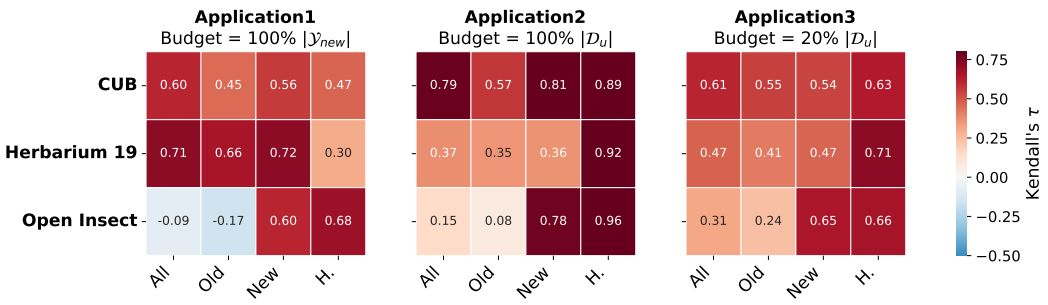

Figure 3: **Kendall's rank correlation τ between clustering quality metrics and downstream performance metrics.** Homogeneity (H.) shows stronger correlations with downstream performance than the three clustering accuracy metrics commonly used in literature.

Configuration Selection. Given the sensitivity to these components, making `Sciurus` useful in practice requires identifying configurations that perform consistently well across use cases. While no single configuration dominates across all use cases (see Section F in the Appendix), two trends emerge: DINOv3 is consistently among the best-performing backbones across all three datasets, and SimGCD appears to be an effective GCD training objective. To identify robust configurations, we compute the average performance gap of each configuration relative to the best-performing configuration across use case and report the configuration with the smallest average gap for each application in Table 3. All three configurations use DINOv3 fine-tuned from block 11 and sample margin as the sorting strategy. We further train SimGCD using the ground-truth number of categories and find that this either has little effect or improves performance, indicating that the performance achieved by `Sciurus` is realistic instead of an artifact of our estimated number of categories. Better estimates could therefore further improve performance (see Section G in the Appendix for details).

How does Open-Set Recognition help? For each configuration listed in Table 3, pre-filtering $\mathcal{D}_u$ based on confidence has little or no impact on downstream performance. This is because all configurations already use sample margin as the sorting strategy, which prioritizes samples with lower margins. Since lower margins are strongly correlated with lower confidence, pre-filtering based on confidence provides limited additional information. Although OSR does not directly improve downstream performance, it serves as the best query strategy for Application 1 and the best sorting strategies for all applications. Moreover, the top four configurations for each application use OSR-based scores (sample or cluster margin, and sample or cluster confidence) rather than cluster variance. However, OSR scores alone are insufficient as all five baseline methods show a big performance gap from `Sciurus`. These results suggest that GCD and OSR should be combined together to achieve good downstream performance.

Table 3: **Configurations with the smallest average performance gap relative to the best-performing configuration for each application.** All selected configurations use DINOv3 fine-tuned from block 11 and sample margin as the sorting strategy.

Application	Method	Query Strategy	Gap
1	SimGCD	Least Margin	1.47
2	SimGCD	Cluster Center	2.13
3	SimGCD k	Cluster Center	1.08

8 Conclusion

By formulating three practical discovery tasks and introducing corresponding evaluation metrics, we demonstrate how GCD can support biodiversity monitoring and greatly outperform alternative sample selection methods. Meanwhile, our results show that making GCD useful in practice is a non-trivial task. The downstream performance of `Sciurus` depends not only on the choice of GCD model, but also on other components of the selection pipeline. Moreover, we show that Open-Set Recognition can provide complementary information to GCD, by guiding the query or sorting strategies. Together, these findings highlight both the potential of GCD for biodiversity monitoring and the importance of designing the surrounding selection pipeline to translate clustering quality into practical gains.

AI USE STATEMENT

In this work, we used generative AI to "implement methods" by using it to write code. The following tasks are not applicable: generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis." We have not used generative AI for other tasks that require disclosure.

Additionally, we used generative AI tools to help write the figure and table captions and to improve the readability of Algorithm 1. All AI-generated text has been proofread or modified by human. LLM-generated code was verified by one author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

For GCD model training, we use existing repositories or packages detailed in Section B. For the selection pipeline, we provide pseudo code in Algorithm 1 and describe the implementation details of each component in Section 5. Code will be made public upon acceptance.

REFERENCES

Lukas Adam, Vojtech Cermak, Kostas Papafitsoros, and Lukas Picek. Wildlifereid-10k: Wildlife re-identification dataset with 10k individual animals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2124–2134, 2025. 1, 3

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*, 2020. URL https://openreview.net/forum?id=ryghZJBKPS. 4, 7

Anthony D Barnosky, Nicholas Matzke, Susumu Tomiya, Guinevere OU Wogan, Brian Swartz, Tiago B Quental, Charles Marshall, Jenny L McGuire, Emily L Lindsey, Kaitlin C Maguire, et al. Has the earth's sixth mass extinction already arrived? *Nature*, 471(7336):51–57, 2011. 1

Sara Beery, Dan Morris, and Siyu Yang. Efficient pipeline for camera trap image review. *arXiv preprint arXiv:1907.06772*, 2019. 3

Tanya Y Berger-Wolf, Daniel I Rubenstein, Charles V Stewart, Jason A Holmberg, Jason Parham, Sreejith Menon, Jonathan Crall, Jon Van Oast, Emre Kiciman, and Lucas Joppa. Wildbook: Crowdsourcing, computer vision, and data science for conservation. *arXiv preprint arXiv:1710.08880*, 2017. 3

Katharine T Bigham and Ada Carter. A machine learning assisted method for rapidly annotating benthic megafauna in large volumes of marine imagery. *Limnology and Oceanography: Methods*, pp. e70057, 2026. 3

Olaf Bánki, Yury Roskov, Markus Döring, Geoff Ower, Diana Raquel Hernández Robles, Camila Andrea Plata Corredor, Thomas Stjernegaard Jeppesen, Ari Örn, Thomas Pape, Donald Hobern, Stephen Garnett, Holly Little, R. Edward DeWalt, Joe Miller, Thomas Orrell, Rolf Aalbu, John Abbott, Carlos Aedo, E. Aescht, Sara Alexander, Miguel A. Alonso-Zarazaga, Belinda Alvarez, Robert Anderson, Giovani Carlos Andrella, Dragan Z. Antić, Lucas Silveira Antonietto, Claudia Arango, Tom Artois, Margoth Atahuachi Burgos, Stephen Atkinson, John J. Atwood, Ângela Lúcia Bagnatori Sartori, Nicolas Bailly, Joaquín Baixeras, Edward Baker, Anoop Balan, Roger Bamber, Subir Bandyopadhyay, Anthony Barber, H. Barber-James, Rafael Barbosa Pinto, Russell Barrett, L. Bartolozzi, I. Bartsch, George Beccaloni, C.L. Bellamy, Denise Bellan-Santini, P.F. Bellinger, Yair Ben-Dov, Isabel Blasco-Costa, James S. Boatwright, Phil Bock, Barry Bolton, Leonardo M. Borges, Roseli Bortoluzzi, Robert L. Bossard, Cornelio Bota-Sierra, Patrice Bouchard, Nicole Boury-Esnault, Rodrigo Bouzan, Geoff Boxshall, Christopher Boyko, Simone Brandão, Holger Braun, Rod Bray, Gunnar Brehm, John C. Brinda, Paul D Brock, Steven L. Broich, John Brown, Samuel Brown, Savatore Brullo, Anne Bruneau, Louise Bush, Thies Büscher, Magdalena Błażewicz-Paszkowycz, Analyn Cabras, Stephen Cairns, Michael Calonje, Ivanklin Soares Campos-Filho, Warren Cardinal-McTeague, Domingos Cardoso, Lilian Cardoso, Ana Beatriz Gomes de Linz Carvalho, R.C. Castilho, Elizeu B. Castro, Isabella C. Castro Silva, Angélica Cervantes, Alexey Chernyshev, Herve Chevillotte, Le Min Choo, Henry Choong, K.A. Christiansen, F. Cianferoni, Maria Marta Cigliano, Ruth Clarke, Thiago Cobra e Monteiro, Allen Collins, James Compton, Denis Copilaș-Ciocianu, Laure Corbari, Ralf Cordeiro, CoreoideaSF Team, Kevin Cortés-Hernández, Mark John Costello, Simon Crameri, Jesus A. Cruz-López, Paco Cárdenas, Brian Dagley, Meg Daly, Mikhail Daneliya, Rakhee Dangi, Jean-Claude Dauvin, Peter Davie, Claude De Broyer, Sammy De Grave, Haroldo C. De Lima, Jurate De Prins, Willy De Prins, Filipe De Sousa, Manuel De la Estrella, Rob DeSalle, Peter Decker, Wim Decock, Alfonso Delgado-Salinas, Cyrille Deliry, Pablo M Dellapé, J. Den Heyer, Barbara D. Denno, Klaas-Douwe Dijkstra, D.A. Dmitriev, Martin Dohrmann, Óscar Dorado, Franck Dorkeld, Rachel Downey, Lei Duan, Rodrigo Duno de Stefano, Maria-Cristina Díaz, David C Eades, Ashley N. Egan, Michael Eitel, Aliya El Nagar, Marc Eleaume, Christian C. Emig, Henrik Enghoff, Patricia Esquete Garrote, G.A. Evans, Neal L Evenhuis, Marcus Falcão, Frank Farruggia, Kristian Fauchald, Daphne Fautin, Colin Favret, Julia Ferm, Blanca Figuerola, Brian Fisher, Cene Fišer, Andrés Fonseca-Cortés, L. Forró, Ana Paula Fortuna-Perez, Helen Fortune-Hopkins, Peter Fritsch, Anne Fuchs, Shinta Fujimoto, Hidetaka Furuya, Edeline Gagnon, Horia Galea, Mayrolin García Morales, Rade Garic, Rebeca Gasca, J.-L. Gattolliat, Sarah Gerken, Alexandre Gibau de Lima, David Gibson, Cees Gielis, Todd Gilligan, Gonzalo Giribet, Jennifer C. Girón Duque, Arjan

Gittenberger, Gianpietro Giusso del Galdo, Serge Gofas, Mikhail Goncharov, Anne Isabelley Gondim, Claire Goodwin, Rafaël Govaerts, Michal Grabowski, Alexia de A. Granado, Bernarda de Souza Gregório, John R. Grehan, Rosaura Grether, Onno Gross, José Manuel Guerra-García, Alberto Guglielmone, Eric Guilbert, Tobias Guldberg Frøslev, Josef Gusenleitner, Markus Göker, Fabian Haas, Kerry A. Hadfield, Eduardo Hajdu, Michael Hassler, Michael W. Hastriter, Axel Hausmann, Bruce W. Hayward, Ed Hendrycks, Derek Hennen, Thomas J Henry, F.A. Hernandes, Héctor Hernández Macías, Juan Carlos Hernández-Crespo, Adrian Hine, Boon-Chuan Ho, Alicia Hodson, Bert Hoeksema, Mario Hoenemann, J. Holstein, Matthew Hooge, John Hooper, Heidi Hopkins, Ivan Horak, Tammy Horton, Jirí Hošek, Colin Hughes, Lauren Hughes, Thomas Hughes, Rony Huys, Jiří Háva, C. Häuser, João Iganci, Luiz Felipe Moretti Iniesta, F. Janssens, Damià Jaume, Firouzeh Javadi, Krzysztof Jazdzewski, Christian D. Jersabek, Kevin P Johnson, Lucas Jordão, Piotr Jóźwiak, Hiroshi Kajihara, Keiichi Kakui, Axel Kallies, Marcin Jan Kamiński, Kojun Kanda, Jeyaraney Kathirithamby, Michelle Kelly, Young-Hyo Kim, Rachael King, Paul Kirk, Ian Kitching, Michelle Klautau, Bente B. Klitgaard, Nikol Kmentova, Stefan Koenemann, Erik Koenen, N.M. Korovchinsky, A. Kotov, Tatiana Kramina, Traudl Krapp-Schickel, Antonina Kremenetskaia, Andreas Kroh, A.S. Kroupa, Adriano B. Kury, Milena S. Kury, Jiří Kvaček, Olivier Lachenaud, Carlos Lado, Gretchen Lambert, Lorena Lana C. Atunes, Matt Lavin, David Lazarus, François Le Coze, Marianne Le Roux, Sara LeCroy, José Ledis Linares, Sangmi Lee, Martin F. Leitner, Gwilym P. Lewis, Shi-Jin Li, Ji Li-Qiang, R.W. Lichtwardt, Swee-Cheng Lim, Dhugal Lindsay, V. Lohrmann, Stuart J. Longhorn, Wolfgang Lorenz, Jim Lowry, Federico Lozano, Ryan Lumen, Chris HC Lyal, Anne-Nina Lörz, Philippe Magnien, Christopher Mah, Noel Mal, Valéry Malécot, Tomasz Mamos, Renata Manconi, Vidal Mansano, Kelly Markello, Koen Martens, Jon H. Martin, Patrick Martin, Kagiso S. Mashego, Svetlana Maslakova, Bruce Maslin, Sawai Mattapha, Maciej Mańko, Catherine McFadden, S. McKamey, J.A. McMurtry, Jackson Means, Miguel Angel Medrano, Jan Mees, Jan P. Meier-Kolthoff, Kenneth Meland, Amanda C. Mendes, Kelly Merrin, N.C. Mesa, Charles Messing, Carlos G.C. Mielke, Alain Migeon, Douglass R. Miller, Gary L. Miller, Claudia Mills, Alessandro Minelli, Lucília Souza Miranda, David Mitchell, MolluscaBase eds., Tina Molodtsova, José Francisco Montenegro Valls, Richard Mooi, André Morandini, Rosana Moreira da Rocha, Christine Morrow, Annah Moteetee, Leidys Murillo-Ramos, Bruce Murphy, J.P.Z. Narita, Nemys eds., U Neu-Becker, Birger Neuhaus, Alfred Newton, Peter Ng Kee Lin, Anh Nguyen, David Nicolson, John E. Nielsen, Ard Nijhof, Teruaki Nishikawa, Jon Norenburg, Shawn A. O'Donnell, Tim O'Hara, R. Ochoa, Hiroyoshi Ohashi, Kazuaki Ohashi, Justin Ollerenshaw, Pjotr Oosterbroek, Dennis Opresko, Roy Osborne, Hans-Jürgen Osigus, J.D. Oswald, Yuzo Ota, Daniel Otte, David Ouvrard, Luciano Paganucci de Queiroz, Arun Pandey, Aidan C. Parte, Gustav Paulay, Dennis Paulson, R. Toby Pennington, Julia da Silva Pereira, Sávio Gabriel Gomes Pereira, Daniel Perez-Gelabert, A. Petrusek, Pete Phillipson, Daniela Pica, Ulisses Pinheiro, Marli Pires Morim, Andrzej Pisera, Brian Pitkin, David Plotkin, Bruna Poatskievick Pierezan, Gary Poore, Maria Povydysh, Erin Powell, Rayran Araújo Praxedes, W.J. Pulawski, Richard Pyle, Franz Pühringer, Hossein Rajaei, Nivohenintsoa Rakotonirina, Gustavo Ramos, Juliana Rando, Fabiana Ranzato Filardi, Lauren Raz, Geoffrey Read, Tony Rees, Mike Reich, James Davis Reimer, Lorenz C. Reimer, Jan Ove Rein, John Reynolds, Judy Rincón, Marc Rius, Tim Robertson, Gaden Robinson, Gaden S (†) Robinson, Estefania Rodríguez, Michael Ruggiero, Pilar Ríos, Lara Milena Sabater, Romain Sabroux, Kaveh Samimi-Namin, A. Sanborn, Munivenkatappa Sanjappa, Arnoldo Santos-Guerra, Joaquim Sardà Carbasse, M. Sartori, Klaus Sattler, Bernd Schierwater, Steve Schilling, Rowan Schley, C. Schmid-Egger, A. Schmidt-Rhaesa, Paul Schoolmeesters, Martin Schorr, Brian Schrire, Peter Schuchert, R.T. Schuh, Christine Schönberg, Rodrigo Schütz Rodrigues, Malcolm Scoble, Guillermo Seijo, Elidiene Priscila Seleme, André Senna, Cristiana Serejo, A. Sforzi, Noa Shenkar, Gustavo Shimizu, Volker Siegel, Petra Sierwald, Pasi Sihvonen, Andréia Silva Flores, Catarina Silva de Carvalho, Elizabeth Simmons, Marcelo Fragomeni Simon, Thomas Simonsen, Charles E. Simpson, Frederic Sinniger, Yotsawate Sirichamorn, Michael Skvarla, Aaron D. Smith, Vincent S Smith, Danilo Soares Gissi, Dmitry Sokoloff, Solange Sotuyo, Guilherme Sousa da Silva, Jesser F. Souza-Filho, Lauren Spearman, Jörg Spelda, A. Steiner, Torben Stemme, Wolfgang Sterrer, Dennis Stevenson, Martin B D Stiewe, Charles H. Stirton, Shannon Straub, G Stueber, Sabine Stöhr, Shweta Subramaniam, Billie Swalla, Jacek Swedo, Heitor Almeida Batista Sá, Nuria Sánchez Santos, M. Sánchez-Ruiz, Martin Vinther Sørensen, D.M. Takiya, Anne Helene Tandberg, Gerard Tavakilian, Kristian Taylor, A. Thessen, James Darwin Thomas, Phillip Thomas, Scott Thomson, ThripsWiki, Erik Thuesen, Mats Thulin, Mike Thurston, Ben Thuy, Antonio Todaro, Benjamin M. Torke, Sz-Yi Tsai, M. Turiault, John R.G. Turner, Thomas Turner, Xavier Turon, Seth Tyler, UCD Community, Peter Uetz, Jonah

M. Ulmer, Jean Vacelet, Daniel Vachard, Wim Vader, Georgina Valls Domedel, Xander Van der Burgt, Leen Vandepitte, Bart Vanhoorne, Mohammad Vatanparast, Tristan Verhoeff, Ronald Vonk, Risto Väinölä, Genefor Walker-Smith, T. Chad Walter, Nina Wambiji, Dominic Wanke, Les Watling, Haylee Weaver, J. Webb, W.C. Welbourn, Thomas Wesener, Christopher Whipps, Kristine White, Nicholas Wilding, Gary Williams, Annette Wilson, Annette J.G. Wilson, Peter Wing, Sophie Winitsky, Christopher C. Wirth, Martin Wojciechowski, Simon Woodman, Joana Xavier, Tingshuang Yi, Mathew Yoder, Dicky Sick Ki Yu, Nikolai Yunakov, J. Zahniser, Wolfgang Zeidler, Rong Zhang, F. Zinetti, Jean-Loup d'Hondt, G.J. de Moraes, Ana Beatriz Ramos de Oliveira, Nicole de Voogd, M. Guadalupe del Río, Ton van Haaren, E.J. van Nieukerken, Leen van Ofwegen, Rob van Soest, Ozan Şentürk, ITIS, International Committee on Taxonomy of Viruses (ICTV), Legume Phylogeny Working Group (LPWG), Natural History Museum Bern, Paleobiology Database contributors, and World Flora Online. Catalogue of life, 9 2026. ISSN 2405-8858. URL https://www.checklistbank.org/dataset/316321. 1

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020. 4

Yuyan Chen, Nico Lang, B. Schmidt, Aditya Jain, Yves Basset, Sara Beery, Maxim Larrivee, and David Rolnick. Open-insect: Benchmarking open-set recognition of novel species in biodiversity monitoring. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-1859. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5048dada2e09b756fc139fa810cece30-Paper-Datasets_and_Benchmarks_Track.pdf. 4, 6

Robert H Cowie, Philippe Bouchet, and Benoît Fontaine. The sixth mass extinction: fact, fiction or speculation? *Biological Reviews*, 97(2):640–663, 2022. 1

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 2632–2652, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/0b408293619f725fd30162af057e531a-Paper-Conference.pdf. 5

Mikel M Delgado and Lucia F Jacobs. Caching for where and what: evidence for a mnemonic strategy in a scatter-hoarder. *Royal Society open science*, 4(9):170958, 2017. 1

Jianyang Gu, Samuel Stevens, Elizabeth G Campolongo, Matthew J Thompson, Net Zhang, Jiaman Wu, Andrei Kopanev, Zheda Mai, Alexander E. White, James Balhoff, Wasila Dahdul, Daniel Rubenstein, Hilmar Lapp, Tanya Berger-Wolf, Wei-Lun Chao, and Yu Su. Bioclip 2: Emergent properties from scaling hierarchical contrastive learning, 2025. URL https://arxiv.org/abs/2505.23883. 1, 3

Jizhou Han, Shaokun Wang, Yuhang He, Chenhao Ding, Qiang Wang, Xinyuan Gao, SongLin Dong, and Yihong Gong. Consistent supervised-unsupervised alignment for generalized category discovery. *arXiv preprint arXiv:2507.04725*, 2025. 4

Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8401–8409, 2019. 4

Zhenqi He, Yuanpei Liu, and Kai Han. Seal: Semantic-aware hierarchical learning for generalized category discovery. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 160765–160794. Curran Associates, Inc., 2025. doi: 10.52202/085713-5369. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/eb3c42ddfa16d8421fdba13528107cc1-Paper-Conference.pdf. 2, 4, 5, 16

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 4

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 19

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4

Yuanpei Liu, Zhenqi He, and Kai Han. Hyperbolic category discovery. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9891–9900, 2025. 4, 5

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*, pp. 16890–16900, 2024. 4

B. E. Moore and J. J. Corso. Fiftyone. *GitHub. Note: https://github.com/voxel51/fiftyone*, 2020. 3

Camilo Mora, Derek P Tittensor, Sina Adl, Alastair GB Simpson, and Boris Worm. How many species are there on Earth and in the ocean? *PLoS biology*, 9(8):e1001127, 2011. 1

Margarita Mulero-Pázmány, Sandro Hurtado, Cristóbal Barba-González, María Luisa Antequera-Gómez, Francisco Díaz-Ruiz, Raimundo Real, Ismael Navas-Delgado, and José F Aldana-Montes. Addressing significant challenges for animal detection in camera trap images: a novel deep learning-based approach. *Scientific Reports*, 15(1):16191, 2025. 1

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 5

Sarah Rastegar, Mohammadreza Salehi, Yuki M Asano, Hazel Doughty, and Cees GM Snoek. Selex: Self-expertise in fine-grained generalized category discovery. In *European Conference on Computer Vision*, pp. 440–458. Springer, 2024. 4, 5

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW. 4, 7

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, 2025. URL https://arxiv.org/abs/2508.10104. 5

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, et al. Bioclip: A vision foundation model for the tree of life. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19412–19424, 2024. 3, 5

Asger Svenning, Guillaume Mougeot, Jamie Alison, Daphne Chevalier, Nisa Chavez Molina, Song-Quan Ong, Kim Bjerge, Juli Carrillo, Toke Thomas Høye, and Quentin Geissmann. A general method for detection and segmentation of terrestrial arthropods in images. *Methods in Ecology and Evolution*, 17(3):727–739, 2026. 3

Hubert A Szczygieł, Brianna Johns, Bernat Fortet, Daisy H Dent, Kit Quitmeyer, and Andrew Quitmeyer. Mothbox and mothbot: automated light trap and data processing system for scalable insect monitoring. *Methods in Ecology and Evolution*, 17(7):2082–2091, 2026. 3

Kiat Chuan Tan, Yulong Liu, Barbara Ambrose, Melissa Tulig, and Serge J. Belongie. The herbarium challenge 2019 dataset. *CoRR*, abs/1906.05372, 2019. URL http://arxiv.org/abs/1906.05372. 2, 4

Daniel Thornton, Dan Morris, Travis King, Lucy Perera-Romero, Alissa Anderson, Rony Garcia-Anleu, Scott Fitkin, and Carly Vynne. Identification of camera trap images by artificial intelligence and human experts produces similar multi-species occupancy models. *Journal of Applied Ecology*, 63(5):e70370, 2026. doi: https://doi.org/10.1111/1365-2664.70370. URL https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/1365-2664.70370. e70370 JAPPL-2025-02072.R1. 1

Devis Tuia, Benjamin Kellenberger, Sara Beery, Blair R Costelloe, Silvia Zuffi, Benjamin Risse, Alexander Mathis, Mackenzie W Mathis, Frank Van Langevelde, Tilo Burghardt, et al. Perspectives in machine learning for wildlife conservation. *Nature communications*, 13(1):1–15, 2022. 1

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7492–7501, 2022. 1, 4

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 4

Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 112–119, 2014. doi: 10.1109/IJCNN.2014.6889457. 4, 7

Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16590–16600, 2023. 4, 5

A ESTIMATE THE NUMBER OF CATEGORIES

In Table 4, we show the estimated number of categories in the unlabeled dataset and in both unlabeled and labeled datasets using agglomerative clustering as discussed in Section 5.1. All backbones achieve similar results and hence. Thus, for simplicity, we use the estimates by DINOv3 for all subsequent experiments.

Table 4: Estimation of $|\mathcal{Y}_u|$ and $|\mathcal{Y}_l \cup \mathcal{Y}_u|$ by three off-the-shelf foundation models.

Dataset	Ground Truth		BioCLIP			DINOv2			DINOv3		
	k_{all}	k_{u}	$\hat{k}_{\text{all}}$	$\hat{k}_{\text{u}}$	τ	$\hat{k}_{\text{all}}$	$\hat{k}_{\text{u}}$	τ	$\hat{k}_{\text{all}}$	$\hat{k}_{\text{u}}$	τ
CUB	200	200	221	220	17	254	254	41	228	228	34
Herbarium 19	683	683	1213	1213	13	1153	1512	35	1139	1139	23
Open Insect	1771	1770	1846	1846	19	1818	1816	45	1803	1802	33

In Table 5, we show the estimated number of categories in $\mathcal{D}_l \cup \mathcal{D}_u$ at each level for SEAL. Each level corresponds to a taxonomic hierarchy. Level 0 indicates "Species" for all datasets. Following He et al. (2025), we use "Family" as Level 1 for CUB and "Genus" as Level 1 for Herbarium 19. We use "Genus" as Level 1 for Open Insect as well.

Table 5: Estimate the number of categories at different levels for SEAL using DINOv3.

Dataset	Level 0			Level 1		
	GT	Estimate	Threshold	GT	Estimate	Threshold
CUB	200	228	34	38	94	67
Herbarium	683	1139	23	63	190	50
Open-Insect	1771	1803	33	813	1142	40

B TRAINING DETAILS

For SimGCD[2], SEAL[3], and HypSelEx[4], we use the official code implementation (see footnote). For Agglomerative Clustering and K-Means, we use the scikit-learn implementation and the random seed of K-Means is fixed as 42. All training can be done on 1 RTX8000 GPU with 8 CPUs and 64G CPU memory in total. For Open Insect, Agglomerative Clustering may require up to 96G CPU memory.

[2] https://github.com/CVMI-Lab/SimGCD
[3] https://github.com/Visual-AI/seal
[4] https://github.com/Visual-AI/HypCD

C APPLICATION 2

For Application 2, we compare methods based on their Pareto dominance. A method is Pareto-optimal if no other method achieves both higher discovery rate and higher efficiency. As shown in Figure 4, for all budgets and datasets, `Sciurus` is Pareto-optimal compared to other methods.

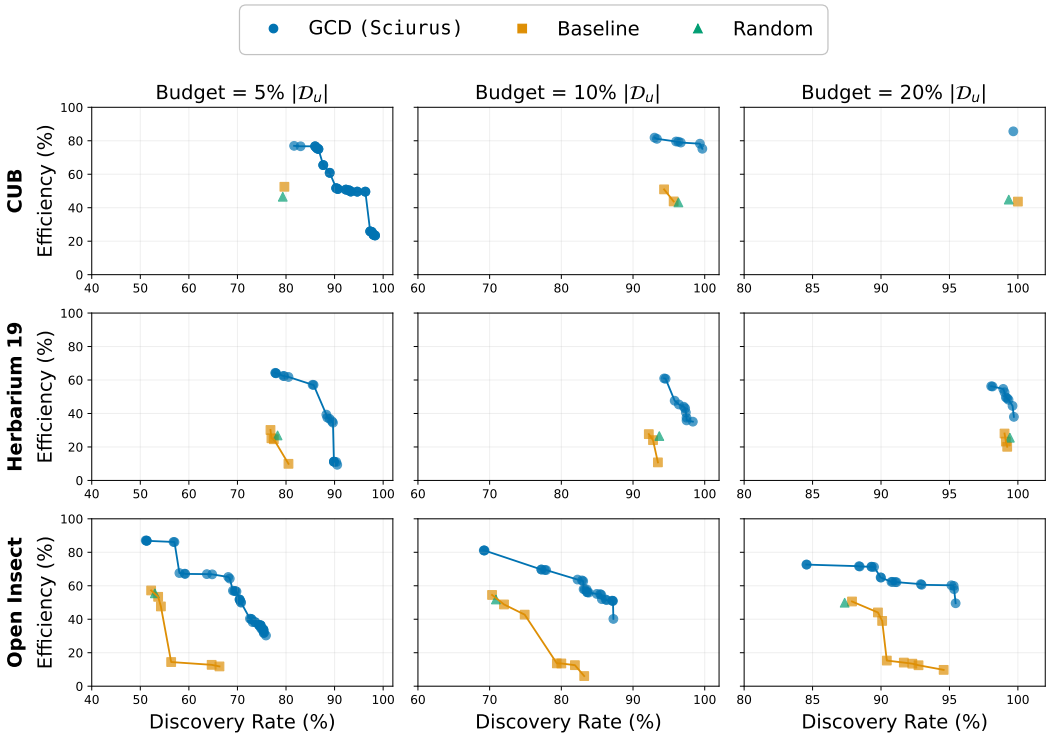

Figure 4: **Selection Method Comparison for Application 2.** For each dataset and budget, we show only the Pareto-optimal points among `Sciurus` and the baseline selection methods, where no other method achieves both higher discovery rate and higher efficiency. Non-Pareto-optimal points are omitted for clarity.

.

We did an ablation study to evaluate the effect of within-cluster sorting strategies. For each dataset, budget, query strategy, GCD method, and backbone, we compute the difference between the maximum and minimum *Efficiency* obtained using the three within-cluster sorting strategies: *Sample Confidence*, *Sample Margin*, and *Distance to Cluster Center*. We then average these differences across query strategies, GCD methods, and backbones (see Table 6)

Table 6: **Variation in annotation efficiency across within-cluster sorting strategies.** For each dataset, budget, query strategy, GCD method, and backbone, we compute the difference between the maximum and minimum *Efficiency* obtained using the three within-cluster sorting strategies: *Sample Confidence*, *Sample Margin*, and *Distance to Cluster Center*. We then average these differences across query strategies, GCD methods, and backbones.

Budget	Dataset	Mean	Std
5	CUB	0.09	0.21
10	CUB	0.77	0.69
20	CUB	1.21	0.87
100	CUB	2.90	2.05
5	Herbarium 19	0.30	0.46
10	Herbarium 19	0.65	0.66
20	Herbarium 19	0.99	0.60
100	Herbarium 19	0.90	0.70
5	Open Insect	1.15	1.46
10	Open Insect	1.92	1.64
20	Open Insect	2.29	1.43
100	Open Insect	3.94	1.63

D APPLICATION 3

In Figure 2, we report test accuracy using nearest-neighbor classification. We did an ablation study to evaluate the effect of linear probing. Specifically, for each dataset and budget, we train a linear classifier on the extracted features for 10 epochs using Adam (Kingma & Ba, 2014) with a learning rate of 0.1. For each dataset, budget, query strategy, GCD method, and backbone, we compute the difference between the nearest-neighbor classifier and linear probing. We then average these differences across query strategies, GCD methods, and backbones (see Table 7)

Table 7: **Variation in test accuracy across classification methods.** For each dataset, budget, query strategy, GCD method, and backbone, we compute the difference between the nearest-neighbor classifier and linear probing. We then average these differences across query strategies, GCD methods, and backbones.

Budget	Dataset	Mean	Std
5	CUB	6.49	4.07
10	CUB	5.03	3.57
20	CUB	3.74	2.35
5	Herbarium 19	5.22	3.55
10	Herbarium 19	5.00	3.03
20	Herbarium 19	4.56	3.08
5	Open Insect	7.94	2.72
10	Open Insect	6.70	1.93
20	Open Insect	5.19	1.90

E METRIC CORRELATION

In Section 6, we showed that `Sciurus`, our GCD-based selection pipeline, can substantially improve downstream performance compared with other selection methods. However, this analysis focused only on the best-performing GCD configuration. This raises a natural question: do different GCD methods consistently provide similar downstream performance, or does the choice of GCD method matter? To investigate this, we measure the performance gap across the seven GCD methods using 1) the best performing query and sorting strategies (shown in Figure 5) and 2) under fixed query and sorting strategies (shown in Figure 6). Under both scenarios, we observe a non-negligible gap in downstream performance, suggesting that the choice of GCD methods is important for downstream tasks.

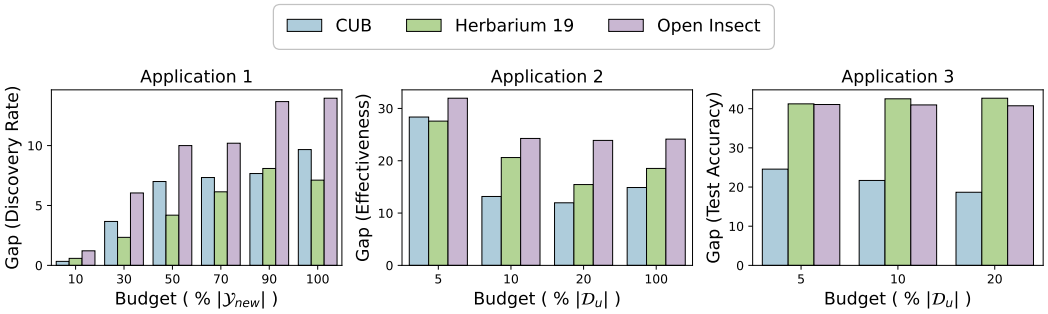

Figure 5: **Performance gap across GCD methods using their best-performing query and sorting strategies.** For each dataset, budget, and application, we compute the gap between the highest- and lowest-performing GCD methods after selecting the best query and sorting strategy for each method.

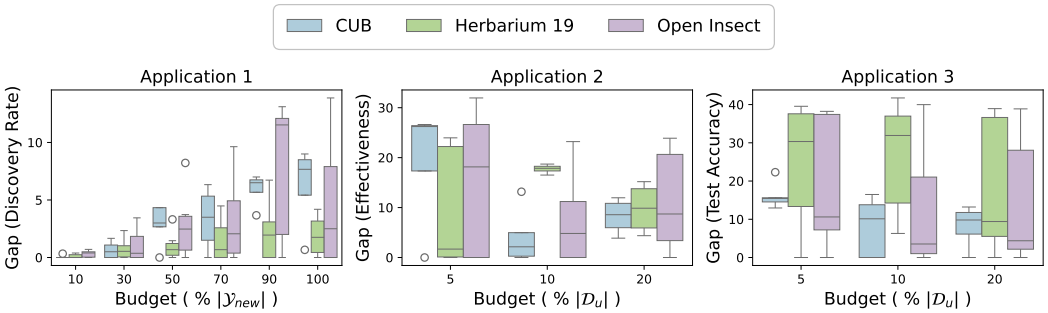

Figure 6: **Distribution of performance gaps across GCD methods under fixed query and sorting strategies.** For each dataset, budget, and application, we compute the performance gap between the highest- and lowest-performing GCD methods for each query and sorting strategy combination. Boxplots show the distribution of these gaps across strategy combinations. The gap between GCD methods are still non-negligible even when query and sorting strategies are held fixed.

Relevance of Clustering Output to Downstream Performance. For Application 2 at budgets below 100%, the relationship is more difficult to interpret because downstream performance reflects a trade-off between discovery rate and verification efficiency. In particular, a method may appear less efficient simply because its clustering enables the selection of a larger number of categories for verification, making clustering quality and downstream performance difficult to disentangle.

For Application 3 at a 100% budget, the clustering assignments do not affect the downstream task, since all samples are available and GCD only contributes through the learned representation used for nearest-neighbor prediction. Consequently, clustering quality is irrelevant in this setting.

Why is Homogeneity better correlated to downstream tasks?

For example, consider two categories A and B with 100 samples each and the cluster assignments in Tables 8 and 9. Both have $\mathrm{ACC}(\mathrm{All}) = 160/200 = 0.8$, but their Homogeneity scores are 1 and

0.825, respectively. This difference is relevant downstream: Assignment 1 guarantees a Discovery Rate of 1 with a budget ≥ 3 and requires ≤ 2 switches, whereas Assignment 2 provides no such guarantees and may require substantially more switches. Thus, Homogeneity can capture differences in clustering structure that ACC overlooks but that are relevant to our downstream tasks.

Table 8: Cluster assignment 1

	C_1^1	C_1^2	C_1^3
A	40	60	0
B	0	0	100

Table 9: Cluster assignment 1

	C_2^1	C_2^2	C_2^3
A	80	10	10
B	10	80	10

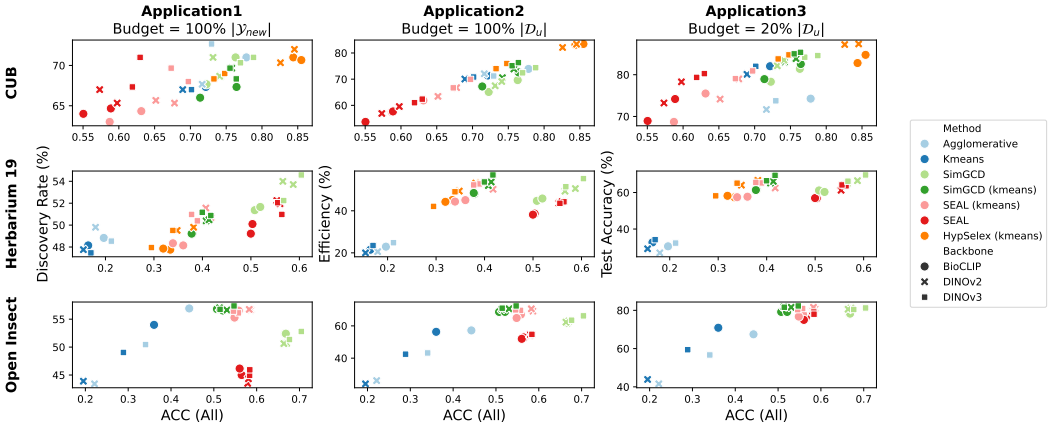

Figure 7: **Scatterplot of Discovery Rate vs ACC (All).** For methods that require training, each method has 6 data points representing 6 combinations of 2 fine-tuning strategies (from block 11 or from blocks 10 and 11) and 3 backbones. We do not differentiate the fine-tuning strategies in the plot for better presentation. For method that do not training, each method has 3 points indicating 3 backbones.

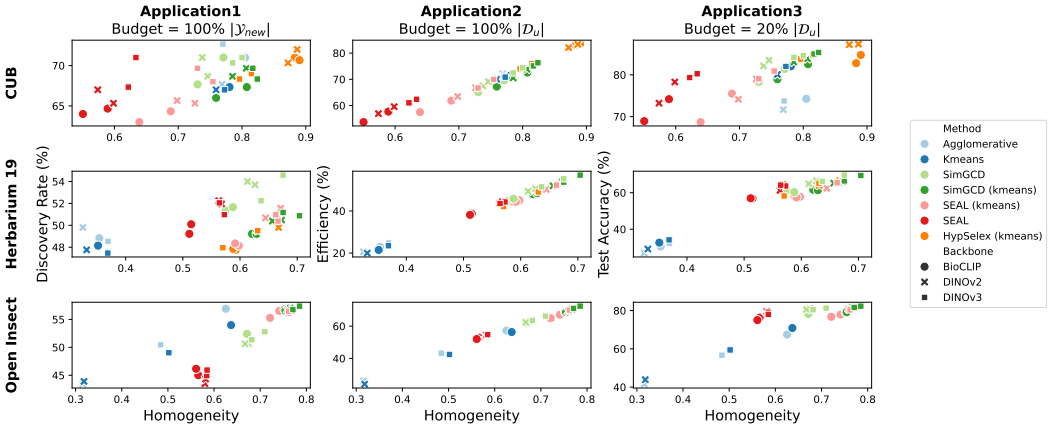

Figure 8: **Scatterplot of Discovery Rate vs Homogeneity.** For methods that require training, each method has 6 data points representing 6 combinations of 2 fine-tuning strategies (from block 11 or from blocks 10 and 11) and 3 backbones. We do not differentiate the fine-tuning strategies in the plot for better presentation. For method that do not training, each method has 3 points indicating 3 backbones.

F CONFIGURATION SELECTION

Table 13 shows that no single configuration rules all. Nevertheless, two trends emerge. First, DINOv3 is consistently among the best-performing backbones across all three datasets, suggesting that it provides a robust feature representation across applications. Second, SimGCD appears to be an effective GCD training objective. However, there is no clear consensus on how to best leverage its output: in some use cases, directly clustering the predictions from its classification head yields the best performance, whereas in others, using SimGCD as a feature extractor followed by kmeans clustering performs better.

To identify configurations that perform consistently well, we measure the performance gap of each configuration relative to the best-performing configuration for each application, budget, and dataset. For each application , we average the gaps across all budget and datasets and rank configurations by this average gap. We show the five configurations with the smallest gap in Table 13.

Table 10: **Application 1: Discovery Rate.** For each dataset and budget, we show the configuration that achieves the highest mean discovery rate across three runs with the standard deviation shown in the parentheses.

Dataset	Budget	Method	Gradient Block	Inference	Backbone	Query Strategy	Sorting Strategy	Discovery Rate
CUB	10	Agglomerative	-	-	BioCLIP	Cluster Center	Cluster Confidence	10.00 (0.00)
CUB	30	Agglomerative	-	-	BioCLIP	Least Confidence	Cluster Confidence	29.00 (0.00)
CUB	50	Agglomerative	-	-	BioCLIP	Cluster Center	Cluster Margin	45.33 (0.58)
CUB	70	Agglomerative	-	-	BioCLIP	Cluster Center	Cluster Confidence	57.33 (1.15)
CUB	90	Agglomerative	-	-	DINOv3	Cluster Center	Sample Confidence	67.67 (0.58)
CUB	100	Agglomerative	-	-	DINOv3	Cluster Center	Cluster Margin	72.67 (1.53)
Herbarium 19	10	SEAL	10	cls	DINOv2	Least Confidence	Sample Confidence	8.77 (0.58)
Herbarium 19	30	SEAL	11	cls	DINOv3	Least Margin	Sample Margin	22.51 (0.58)
Herbarium 19	50	SimGCD	11	cls	DINOv2	Least Margin	Sample Margin	33.82 (0.17)
Herbarium 19	70	SimGCD	11	cls	DINOv3	Least Margin	Sample Margin	43.37 (1.03)
Herbarium 19	90	SimGCD	11	cls	DINOv3	Least Margin	Sample Margin	51.75 (1.83)
Herbarium 19	100	SimGCD	11	cls	DINOv3	Least Margin	Sample Margin	54.58 (1.61)
Open Insect	10	SimGCD	10	cls	BioCLIP	Cluster Center	Sample Confidence	9.69 (0.21)
Open Insect	30	SimGCD	11	cls	DINOv3	Cluster Center	Sample Confidence	27.09 (0.64)
Open Insect	50	SimGCD	11	cls	DINOv3	Least Confidence	Cluster Confidence	38.87 (1.24)
Open Insect	70	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Confidence	46.38 (0.26)
Open Insect	90	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Confidence	54.35 (0.43)
Open Insect	100	SimGCD	11	kmeans	DINOv3	Least Margin	Cluster Confidence	57.37 (0.67)

Table 11: **Application 2: Effectiveness.** For each dataset and budget, we show the configuration that achieves the highest mean discovery rate across three runs with the standard deviation shown in the parentheses.

Dataset	Budget	Method	Gradient Block	Inference	Backbone	Query Strategy	Sorting Strategy	Effectiveness
CUB	5	SEAL	11	cls	DINOv3	Cluster Center	Sample Confidence	81.35 (1.73)
CUB	10	HypSelEx	10	kmeans	BioCLIP	Cluster Center	Cluster Confidence	88.80 (0.40)
CUB	20	HypSelEx	11	kmeans	BioCLIP	Cluster Center	Cluster Confidence	92.65 (1.19)
CUB	100	HypSelEx	11	kmeans	BioCLIP	Full Dataset	Full Dataset	91.84 (0.34)
Herbarium 19	5	SimGCD	11	cls	DINOv3	Cluster Center	Sample Confidence	71.38 (0.26)
Herbarium 19	10	SimGCD	11	cls	DINOv3	Cluster Center	Sample Margin	77.64 (0.52)
Herbarium 19	20	SimGCD	11	cls	DINOv3	Cluster Center	Sample Confidence	77.16 (0.31)
Herbarium 19	100	SimGCD	11	kmeans	DINOv3	Full Dataset	Full Dataset	78.59 (0.33)
Open Insect	5	SimGCD	11	cls	DINOv3	Cluster Center	Sample Confidence	71.57 (0.26)
Open Insect	10	SimGCD	11	cls	DINOv3	Cluster Center	Sample Margin	75.18 (0.16)
Open Insect	20	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Confidence	80.36 (0.36)
Open Insect	100	SimGCD	11	kmeans	DINOv3	Full Dataset	Full Dataset	86.60 (0.04)

Table 12: **Application 3: Test Accuracy.** For each dataset and budget, we show the configuration that achieves the highest mean discovery rate across three runs with the standard deviation shown in the parentheses.

Dataset	Budget	Method	Gradient Block	Inference	Backbone	Query Strategy	Sorting Strategy	Test Acc
CUB	5	HypSelEx	11	kmeans	DINOv2	Cluster Center	Cluster Feature Variance	86.42 (0.40)
CUB	10	HypSelEx	11	kmeans	DINOv2	Cluster Center	Sample Confidence	86.87 (0.25)
CUB	20	HypSelEx	10	kmeans	DINOv2	Cluster Center	Cluster Feature Variance	87.37 (0.35)
CUB	100	SimGCD	11	cls	DINOv3	Full Dataset	Full Dataset	87.71 (0.14)
Herbarium 19	5	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Margin	66.25 (0.51)
Herbarium 19	10	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Confidence	68.33 (0.37)
Herbarium 19	20	SimGCD	11	cls	DINOv3	Cluster Center	Cluster Confidence	69.63 (0.56)
Herbarium 19	100	SimGCD	11	cls	DINOv3	Full Dataset	Full Dataset	72.21 (0.50)
Open Insect	5	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Confidence	79.89 (0.09)
Open Insect	10	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Confidence	81.12 (0.08)
Open Insect	20	SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Margin	82.31 (0.01)
Open Insect	100	SimGCD	11	cls	DINOv3	Full Dataset	Full Dataset	84.89 (0.16)

Table 13: **Performance Gap.** For each application, dataset, and budget, we compute the performance gap between each configuration and the best-performing configuration. For each application, we show the five configurations with the smallest "Gap from Best", averaged across three seeds, datasets, and budgets.

Method	Gradient Block	Inference	Backbone	Query Strategy	Sorting Strategy	Application	Gap From Best
SimGCD	11	cls	DINOv3	Least Margin	Sample Margin	Application 1	1.47
SEAL	10	kmeans	DINOv3	Least Margin	Sample Margin	Application 1	2.43
SimGCD	10	cls	DINOv2	Least Margin	Sample Margin	Application 1	2.50
SimGCD	11	cls	BioCLIP	Least Margin	Sample Margin	Application 1	2.54
SimGCD	11	cls	DINOv3	Least Confidence	Sample Confidence	Application 1	2.63
SimGCD	11	cls	DINOv3	Cluster Center	Sample Margin	Application 2	2.13
SimGCD	11	cls	DINOv3	Cluster Center	Cluster Confidence	Application 2	2.17
SimGCD	11	cls	DINOv3	Cluster Center	Sample Confidence	Application 2	2.17
SimGCD	11	cls	DINOv3	Cluster Center	Cluster Margin	Application 2	2.29
SimGCD	11	cls	DINOv3	Cluster Center	Cluster Feature Variance	Application 2	2.90
SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Margin	Application 3	1.08
SimGCD	11	kmeans	DINOv3	Cluster Center	Sample Confidence	Application 3	1.10
SimGCD	11	kmeans	DINOv3	Cluster Center	Cluster Margin	Application 3	1.13
SimGCD	11	kmeans	DINOv3	Cluster Center	Cluster Confidence	Application 3	1.15
SimGCD	11	kmeans	DINOv3	Cluster Center	Cluster Feature Variance	Application 3	1.17

G RESULTS WITH GROUND TRUTH K

Table 14: **Application 1: Sensitivity to k.** We report the highest discovery rate achieved by SimGCD with DINOv3 as the backbone fine-tuned from block 11 among various configurations.

Dataset	Budget	SimGCD	SimGCD (GT k)	Gap
CUB	10	10.00	10.00	0.00
CUB	30	28.00	27.00	1.00
CUB	50	43.00	44.00	-1.00
CUB	70	58.00	58.00	0.00
CUB	90	69.00	68.00	1.00
CUB	100	73.00	72.00	1.00
Herbarium 19	10	9.36	8.48	0.88
Herbarium 19	30	22.51	21.93	0.58
Herbarium 19	50	34.21	33.04	1.17
Herbarium 19	70	44.44	42.11	2.34
Herbarium 19	90	53.80	49.42	4.39
Herbarium 19	100	56.43	53.51	2.92
Open Insect	10	9.66	9.92	-0.25
Open Insect	30	26.36	27.12	-0.76
Open Insect	50	38.05	37.71	0.34
Open Insect	70	43.14	43.31	-0.17
Open Insect	90	49.24	49.41	-0.17
Open Insect	100	51.86	52.20	-0.34
Mean		40.00	39.29	0.72

Table 15: **Application 2: Sensitivity to k.** We report the highest effectiveness achieved by SimGCD with DINOv3 as the backbone fine-tuned from block 11 among various configurations.

dataset	budget	SimGCD	SimGCD (GT k)	Gap
CUB	5	66.74	74.00	-7.26
CUB	10	85.59	87.25	-1.66
CUB	20	88.93	88.56	0.37
CUB	100	86.63	86.00	0.63
Herbarium 19	5	71.19	72.43	-1.23
Herbarium 19	10	78.25	77.07	1.17
Herbarium 19	20	77.58	76.84	0.75
Herbarium 19	100	77.79	76.31	1.48
Open Insect	5	71.73	72.69	-0.96
Open Insect	10	75.15	76.47	-1.32
Open Insect	20	79.31	79.00	0.31
Open Insect	100	83.63	83.90	-0.27
Mean		78.54	79.21	-0.67

Table 16: **Application 3: Sensitivity to k.** We report the highest test accuracy by SimGCD kmeans with DINOv3 as the backbone fine-tuned from block 11 among various configurations.

dataset	budget	SimGCD kmeans	SimGCD kmeans (GT k)	Gap
CUB	5	81.77	82.30	-0.53
CUB	10	83.35	83.15	0.19
CUB	20	84.99	84.32	0.67
Herbarium 19	5	66.37	64.88	1.49
Herbarium 19	10	68.71	66.68	2.03
Herbarium 19	20	69.93	68.90	1.03
Open Insect	5	79.96	79.57	0.39
Open Insect	10	81.05	80.77	0.28
Open Insect	20	82.32	82.05	0.27
Mean		77.61	76.96	0.65