Novelty detection with Active Class-Incremental Learning on Long-tailed datasets

Max M. Lievense
Electrical Engineering, Mathematics and Computer Science
University of Twente
Enschede, Netherlands
max.lievense@outlook.com

Jacob W. Kamminga
Pervasive Systems
University of Twente
Enschede, Netherlands
j.w.kamminga@utwente.nl

***Abstract*—In many practical applications, the effort required for data annotation often exceeds the cost of data acquisition, especially in the presence of long-tailed distributions within an open-set setting. This challenge is further exacerbated when rare classes must be identified in large, unlabeled datasets. Traditional methods operate under the closed-set assumption, which is frequently impractical in real-world scenarios. Existing research on open-set recognition does not fully capture the complexity of discovering novel classes during the annotation process. To address these limitations, we propose a methodology that integrates active learning and class-incremental techniques, utilizing out-of-distribution detection algorithms to efficiently identify novel classes during the annotation process. Our results demonstrate a significant reduction in the annotation effort required to approach a closed-set dataset on three widely used benchmark datasets. Specifically, our methodology discovers *100%* of the classes on Places365-LT and ImageNet-LT with 59.1% and 57.6% fewer annotations, respectively, compared to random sampling, by employing a committee of multiple detectors. Similarly, we discover *99%* of the classes on iNaturalist2018-Plantae with 23.0% fewer annotations.**[1]

I. Introduction

In an increasingly data-driven world, reducing the time and cost associated with annotating large-scale datasets is a critical challenge, particularly in machine learning tasks involving the discovery of classes in complex datasets. A prominent example of this challenge is long-tailed (LT) datasets, characterized by a significant variation in occurrence among different classes. This complex distribution is not just a statistical curiosity; it mirrors real-world scenarios across various domains, such as biodiversity [1], medical diagnosis [2, 3], image segmentation [4], natural language processing, and autonomous vehicles. In these applications, the rare classes are often particularly valuable, making their discovery crucial.

Imagine having a folder containing raw data that you want to use to train a classification model. The first step is to annotate the data, which involves labeling the samples with their desired classes. For example, the folder contains images taken with a wildlife camera, and you want to know the species present in the footage. The folder contains a large number of images of common species, such as deer and squirrels, while rare species, such as wolves, only appear in a handful of images. This LT distribution presents a unique challenge: how do we ensure that all classes, especially the rare ones, are labeled during the annotation process?

Annotating a large-scale LT dataset completely will ensure that all classes are labeled; however, the sheer number of samples makes this impractical and extremely costly. Lowering the cost implies only annotating a portion of the dataset. The LT distribution makes it difficult to ensure that all classes are represented in the labeled dataset. Given that a significant portion of the samples belongs to a small subset of classes, random sampling for annotation would likely focus on the predominant classes, leaving the rare classes underrepresented.

This is where the synergy of Active Learning (AL) [5] with novel class detection can be beneficial. AL aims to select the most valuable samples for annotation on a limited annotation budget. AL methods typically operate under a closed-set scenario where all classes are known prior to training. However, during annotation, the dataset is in an open-set scenario, where unknown/novel classes exist that are not yet represented in the labeled dataset. Novel class detection algorithms can be used to determine the likelihood that unlabelled samples belong to a novel class. This research combines these two approaches and investigates the potential of novelty detection to discover rare classes during the annotation of LT datasets using AL.

The primary objective of this work is to maximize class coverage during training, discovering as many novel classes as possible, while minimizing the number of labels (human effort) required. This focus on class coverage as a key evaluation metric is distinct from the traditional emphasis on classification accuracy. This objective has been used in anomaly detection tasks [6–9] on tabular datasets and very simple image datasets (e.g., GAIT and Digits), but to the best of our knowledge, it has not been applied to the process of annotating long-tailed image datasets.

From a different perspective, the proposed process in this research seeks to ensure that the labeled dataset moves closer to a closed-set assumption, where rare classes are represented more frequently, and a greater number of informative and

[1]Our code, dataset adaptations, and models are publicly available at: github.com/MaxLievense/Active-Class-Incremental-Learning

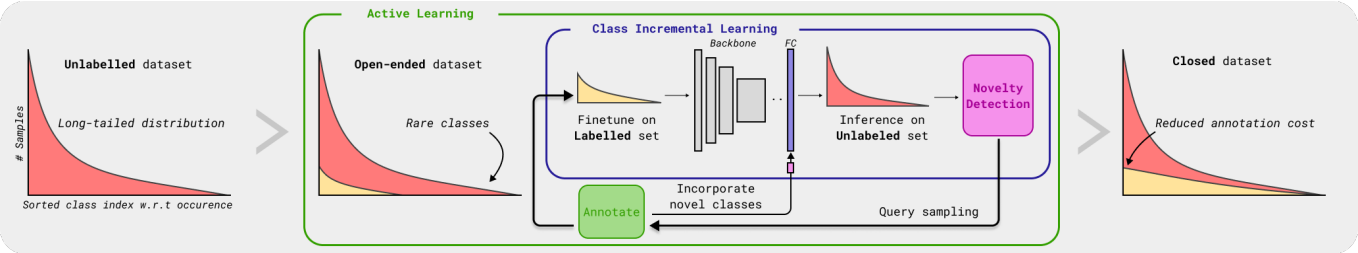

Fig. 1: Efficient long-tailed dataset annotation using active learning, class incremental learning, and novelty detection. Starting from an unlabelled (open-set) dataset, the approach iteratively selects the most valuable samples determined by novelty detection algorithms to discover novel classes and incorporates them in the model. This processes leads to a closed-set dataset with a reduced annotation cost.

diverse samples are labeled. Once sufficiently annotated, the labeled dataset can be utilized to train other classification models that operate under the closed-set assumption, with their primary focus being on improving classification accuracy. Our approach is designed with real-world and practical limitations in mind, making it particularly valuable for application in real-world scenarios beyond controlled research environments.

In summary, the key contributions of this research are:

- **Foundation of a novel research path**: We lay the groundwork for a novel research path focused on maximizing class coverage in long-tailed image datasets. We demonstrate that our approach is distinct from existing open-set recognition and out-of-domain research. We evaluate various training techniques, such as balanced sampling and K-Fold cross-validation, across three unique and widely used long-tailed benchmarking datasets. Additionally, we implement a wide array of novelty detection algorithms, providing a benchmark for future research in this area.
- **Exploration of unsupervised feature representation learning**: We investigate the effectiveness of unsupervised feature representation learning using open-source supervised pre-trained models. While our findings indicate improvements in open-set recognition tasks, the impact on our setting was less pronounced. Despite this, unsupervised models show promise, particularly in scenarios where supervised pre-trained models are inaccessible, thus broadening the applicability of our approach.
- **Substantial reduction in annotation effort**: Using our Active Class-Incremental Learning framework, we demonstrate a significant reduction in the number of annotations required to achieve high class coverage in long-tailed datasets. Specifically, our methodology discovers *100%* of the classes on Places365-LT and ImageNet-LT with 59.1% and 57.6% fewer annotations compared to random sampling, respectively, by employing multiple detectors in a committee approach. Similarly, we discover *99%* of the classes on iNaturalist2018-Plantae with 23.0% fewer annotations.

II. Related Work

Active Learning (AL) traditionally aims to efficiently utilize a limited label budget by selecting the most valuable samples for labeling to maximize the performance of a model [10–16]. In traditional AL, performance is expressed in evaluation accuracy and typically operates under a closed-set assumption, where all desired classes are known prior to training. These implementations focus on sample valuableness, a metric determined by a specific sampling strategy. Although typical AL sampling strategies are not limited to closed-set settings, there is limited research on AL for open-set annotation. In this research, we extend the typical AL sampling strategies to incorporate novelty detection algorithms, which might be better suited for our setting.

A. Novelty Detection

Novelty detection [17] involves identifying and classifying data points that differ from the labeled data available during training. Terms such as anomaly detection and outlier detection are frequently used interchangeably with novelty detection, though they originate from different application domains [18]. Despite the lack of a universally accepted definition, these terms share a common goal: to identify data points that deviate significantly from the norm. Most research in this area focuses on training a model that can subsequently be used to detect novel classes in an open-set scenario, typically as a single-step process. In contrast, our research emphasizes a multi-step methodology, wherein the detection process is repeated during the annotation process, and novel classes are continuously incorporated into the learning process.

A closely related field is Open-Set Recognition (OSR), which focuses on determining whether a given input belongs to any of the known classes, while ensuring accurate classification of these known classes and mitigating false classification of unknown classes. In practice, OSR operates on principles similar to outlier detection at the architectural level, typically adding layers that ensure the confidence level of unknown classes remains below a certain threshold. Various approaches in OSR achieve this by employing techniques such as training additional memory banks to store features of known classes and using clustering distances to adjust classification decisions

[19, 20], or by training binary classifiers to distinguish between known data and outliers [21].

There is a line of research that addresses the ”open-set annotation” challenge. However, in contrast to our research, these works [22–29] assume a closed-set output space. Their approach focuses on selecting examples that belong to closed-set classes for annotation while ignoring unknown ones. Since most of these methods typically operate at the architectural level, they require a substantial labeled dataset to function efficiently. Moreover, incorporating novel classes often necessitates retraining the entire architecture. These techniques are therefore ill-suited for scenarios where the outer layer changes frequently, as is the case in this research.

Another related area is Out-of-Distribution (OOD) detection [30], which focuses on determining whether a sample falls within the training distribution by utilizing anomaly detection algorithms to identify unusual patterns in the data. OOD methods are typically post-hoc or output-level approaches, meaning they analyze the model's outputs without requiring modifications to the model's architecture. This makes OOD detection particularly useful in scenarios where architectural changes occur, such as the incorporation of novel classes during training. This flexibility enables OOD methods to be more easily integrated into existing systems and to address a wider variety of novelty detection challenges. In this research, we categorize a wide variety of OOD detection methods into Probability-based [30], Logit-based [31, 32], Feature-based [33–37], Input-based [38], and Mixed methods [39, 40], as shown in Table I. Each category represents a different approach to novelty detection, offering various strengths depending on the specific settings, elaborated further in Section IV.

OOD techniques are commonly evaluated by appending a pre-defined (open-set) dataset containing classes that are mutually exclusive to the training dataset. However, our research follows the setting where the OOD samples are the novel classes within the unlabeled dataset. This scenario is depicted in Figure 2. Due to the limited number of novel classes within the unlabeled dataset, the complexity of the task is significantly increased, making the evaluation of the detector not directly comparable with other research (as elaborated in Section IV-A). Nevertheless, detector performance can still be assessed using a similar approach, as discussed in Section IV-A. Notably, no prior research has specifically addressed the annotation process for novel classes within the unlabeled dataset, which is the primary focus of our work.

B. Class Incremental Learning

After detecting novel classes, our next step is to incorporate them into the training process, introducing the realm of Class-Incremental Learning (CIL) [41, 42]. This field addresses the open-set problem by allowing the initial dataset to be open-ended and extending the output space during training. One of the primary challenges of CIL is mitigating catastrophic

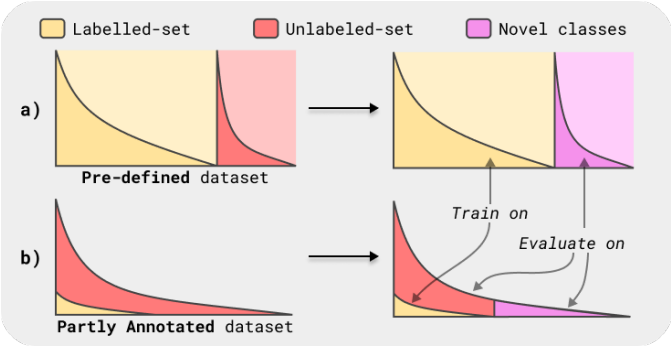

Fig. 2: Illustration of the different definitions of the OOD and our setting's dataset. **a)** Illustrates appending a uniformaly or class imbalanced distributed pre-defined dataset containing mutually exclusive classes from the training set. The OOD evaluate is based on maximizing the accuracy identifying if a sample is from a trained class or novel. **b)** Illustrates our setting, where the dataset is partly annotated and open-set classes are the classes not yet represented in the labeled set. The evaluation is done on the same dataset, where classes can be both known and unknown, and where unknown classes are often the rare classes.

forgetting, where the introduction of new classes causes the model to forget previously learned ones [43]. A common solution for catastrophic forgetting is the use of replay-based techniques [41, 42, 44, 45]. In replay-based techniques, the network revisits former classes by replaying samples from previous tasks, which is the approach we adopt in this research, as discussed in Section III-D1.

In our setting, the likelihood that novel classes belong to the tail classes is high; this aligns closely with the field of Few-Shot Class-Incremental Learning (FSCIL) [41, 42]. FSCIL adapts the N-way K-shot task format from Few-Shot Learning (FSL), which drastically alters the learning process. Furthermore, rather than using traditional CNNs that classify individual samples, FSL and FSCIL often employ meta-learning techniques such as similarity-based networks [46]. Meta-learning techniques are preferred in FSL and FSCIL because they allow the model to learn how to match classes, making it more adaptable to new classes with limited data. Siamese networks use pair-based contrastive loss that attempts to pull the same class closer and push different classes further apart [47, 48], and Prototypical networks use the mean of the embeddings of the samples in a class as the class prototype [49]. Research in FSCIL on large-scale datasets [50, 51] suggests that more refined meta-learning techniques (e.g., class hierarchies) are necessary to decrease classification complexity. This is because the size of the training dataset and the number of classes quickly grow beyond the capabilities of traditional FSL networks, a limitation we are likely to encounter in this research.

C. Unsupervised Feature Representation Learning

Unsupervised feature representation learning [52, 53] has gained significant attention for leveraging large-scale unlabeled datasets, especially in scenarios with limited labeled data. Contrastive learning has proven to be an effective alternative to externally supervised pretrained classification models (which is the de facto standard for pretraining), particularly for domain-specific feature extraction from unlabeled datasets. Studies have shown that unsupervised learning approaches are more robust to class imbalance compared to supervised methods [54], making them advantageous in imbalanced data scenarios. Furthermore, advancements in unsupervised meta-learning frameworks and positive-unlabeled learning strategies have enhanced the ability of models to learn from limited labeled data while efficiently utilizing vast amounts of unlabeled data [55, 56]. These methods collectively contribute to more effective domain-specific feature representation and have demonstrated improved performance across various benchmarks by first learning through unsupervised methods and then fine-tuning on labeled data.

III. Methods

This research explores the performance of various novel class detectors within the field of Out-of-Distribution (OOD) detection, employing an Active Learning (AL) approach. Unlike traditional Open-Set Recognition (OSR) methods, our approach integrates novel classes through Class-Incremental Learning (CIL), allowing the model to continue training on newly discovered classes without the need to retrain the entire model. Thus, this approach works with open-set datasets, where the discovery of new classes dynamically enhances the model's ability to distinguish between them, fostering a continuous learning cycle.

We begin with a small initial labeled dataset and progressively annotate new samples from the unlabeled dataset. This iterative process is particularly novel in the fields of AL, OSR, and CIL, as it leverages the information gained from novel classes to improve further class discovery, creating a feedback loop that refines the model over time.

The primary objective of this research is to identify an efficient method for discovering as many classes as possible during the annotation process while minimizing the number of labels (and therefore human effort) required. Rather than defining what constitutes a "rare class" — a concept that is often subjective and dataset-dependent — our approach emphasizes the ability to discover all classes within an unlabeled long-tailed (LT) dataset.

A. Setting

1) Problem setting: For this research, we designed a setting that emulates a practical annotation scenario on a LT dataset, summarized by the following rules:

- There is no lower limit on the number of samples per class; therefore, a class may exist with only a single sample.
- There is no prior knowledge about the classes at the start of training. This prohibits the use of pre-defined class labels and hierarchies and requires the model to incrementally grow the output space as new classes are discovered.
- A validation dataset is created from the labeled dataset, rather than being provided externally (e.g., a predefined validation set).
- Uniformly distributed testing data is only employed as an evaluation metric in research and does not influence the training process, as this scenario is unlikely in real-world applications.

As stated in Section II, no prior research has been identified that adheres to these criteria, underscoring the novel contribution of this work to the field.

2) Classification setting: Given a dataset $D = D_U \cup D_L$, where D_U denotes unlabeled samples and D_L denotes labeled samples, and $D_U \cap D_L = \emptyset$. In D, there exist N unique classes, and in D_L, there are $N_L \subseteq N$. N_U denotes the novel classes in D_U that are not represented in D_L.

We follow a typical LT classification distribution setting. Given a labeled imbalanced dataset $D_L = \{x_i, y_i\}_{i=1}^{N_L}$ with N_L training classes, where x_i denotes a sample and y_i denotes its label. D_L is ordered by descending in-class samples n_i, meaning $n_i < n_{i+1}$ where n_i is the number of samples in class index i. This creates head classes N_{head} and tail classes N_{tail}, which are the classes with the most and least labeled samples, respectively.

3) Active learning setting: We also follow a typical AL setting. The AL sampling strategy selects unlabeled samples from D_U to be queried by an oracle. In this context, an "oracle" refers to a human annotator or an automated system that provides the correct labels for queried samples. This process is iterative throughout training and expands D_L. There is an initial labeled dataset D_I to which queried samples D_+ are added in each iteration. During the querying process, the overall human effort encompasses the effort required to label both D_I and all D_+. In other words, human effort is equivalent to the total number of labeled samples $|D_L|$, including the initial dataset D_I.

4) Novel class setting: When $D_L \ll D$, due to the LT distribution, it is likely that novel classes N_U exist within D_U. We assume that N_L and N_U are mutually exclusive. During the AL process, novel samples x_+ that belong to a class in N_U are likely to be queried, leading to the discovery of novel classes N_+ while annotating D_+.

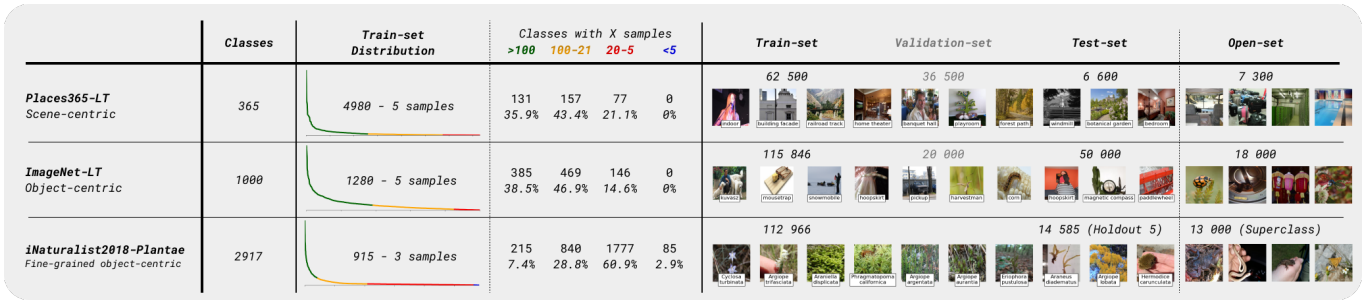

Fig. 3: Overview of the datasets used in the research. Under distribution, the train-set class distribution is sorted with respect to sample occurrence. The line color indicates the tail-category it belongs to, where "> 100" are Head-classes and "$20 - 5$" and "< 5" are Tail-classes. For each subset, the number of samples is shown. The images are sampled from the train-set and open-set. Note that the predefined validation-set is only used in our first experiment.

B. Datasets

In this research, we use three large-scale image classification datasets, as shown in Figure 3. The datasets are:

- **Places365-LT** [19]: A LT version of the original Places365 dataset, containing 1.8 million images from 365 classes. The long-tailed version is a static list of images selected from the original dataset, resulting in a LT distribution. It includes a uniformly distributed test set and a predefined validation set. The open set consists of mutually exclusive classes from all other subsets. All images are of size 256×256 pixels.
- **ImageNet-LT** [19]: A LT version of the original ImageNet2012 dataset, containing 1.2 million images from 1,000 classes. Images are selected similarly to Places365-LT. The images are rescaled to 256×256 pixels, as the dataset contains images of varying sizes.
- **iNaturalist2018-Plantae**: A subset of the iNaturalist2018 [1] dataset, containing only the Plantae superclass. The original dataset has 1.1 million images from 8,142 classes. iNaturalist2018-Plantae is randomly split into two subsets: a training set and a test set, following a holdout rule where a uniform subset of samples is split off from each class. The images are rescaled to 256×256 pixels, as the dataset contains images of varying sizes.

1) Label semantics: Comparing the datasets, Places365-LT is a scene-centric dataset, emphasizing the context and spatial arrangement of multiple objects within a scene, while ImageNet-LT is object-centric, focusing on distinct object features and shapes. Conversely, iNaturalist2018-Plantae is fine-grained, requiring the model to differentiate between subtle variations among plant species. Thus, the chosen datasets offer significant variation in their label semantics, which impacts the model's feature learning, generalization, and may influence novelty detection capabilities.

2) Class distribution: The distributions of Places365-LT and ImageNet-LT are comparable, both having a minimum of 5 samples per class. The main difference lies in the number of samples in the head classes. ImageNet-LT is considered more challenging because it contains three times as many classes as Places365-LT and twice as many total samples. The distribution of iNaturalist2018-Plantae is even more extreme, with over 60% of the classes having fewer than 20 samples. There are nearly three times as many classes to be discovered compared to ImageNet-LT.

Together, these three distinct datasets provide a comprehensive evaluation of the detectors' performance across different label semantics, class distributions, and dataset sizes.

3) Subset creation: Using these datasets, we generate seed-based subsets for use in the research. The dataset is split into the following subsets:

- D_L: Subset containing all labeled samples.
- D_{train}: Class-balanced sampling on a K-Fold cross-validated D_L.
- D_{val}: Validation subset used to determine when to stop training, generated from D_L through K-Fold cross-validation (see Section III-C4). This should not be confused with the predefined validation set.
- D_U: Unlabeled subset on which novelty detection is performed.
- D_{test}: Uniformly distributed subset of all classes, either using the provided external test set or split uniformly from the dataset through a holdout rule.
- D_{open}: An external subset of only open classes used to extend the open-set recognition evaluation, either using the provided external test set or generated from the original dataset.

D_I is created by selecting a configurable number of samples at random to be labeled and used at the start of the training process.

C. Data sampling and manipulation

Our setting requires an unconventional approach to data sampling. Data sampling refers to the process of selecting a subset of data from a larger dataset for training, validation, and testing, and it is typically used in class-imbalanced settings. In our context, data sampling involves selecting from a dynamically growing D_L as it labels samples from D_U. This impacts the class indexes, as novel classes must be added to the output layer of the model. Model validation is done using samples in D_L rather than relying on a predefined validation set. We use K-Fold cross-validation to create D_{train} and D_{val}, which are updated at each epoch. In LT classification and Few-Shot Class-Incremental Learning (FSCIL) tasks, it is common to balance the classes in the training set. Otherwise, head classes are overrepresented compared to tail classes, biasing the model toward head classes and causing it to ignore tail classes. These methods are further elaborated in the following sections.

1) Augmentations: We follow the commonly used ImageNet augmentation, which include random cropping, rotation, flipping, and color adjustments.

2) Annotating data through queries: In our experiments, we use ground-truth datasets but hide the labels where necessary. The implementation can query unlabeled samples from D_U to a virtual oracle, which provides the ground-truth labels and includes them in D_L. This simulates the annotation process of a human. Note that we do not use noisy labels (i.e., incorrect ground-truth labels).

We use a static number of queried samples, denoted as $|D_+|$. In each iteration of the AL process, $|D_+|$ samples are labeled and added to D_L. The samples are selected based on the novelty detection results, further elaborated in Section III-F. The reason for using a static $|D_+|$ is two-fold: (1) it allows for a more controlled evaluation of the detectors, as the number of samples queried is not dependent on detector performance, and (2) it ensures that the annotation process continues, even if the detectors perform poorly.

3) Class reorder: During the creation of D_I, the original class labels are not usable, as this would imply that the output layer of the model contains $[0, 1, 2, ..., |N|]$ as output classes. Firstly, this requires knowledge of the number of classes in the dataset, which is not available in this setting. Secondly, if the open classes exist in the output layer without corresponding samples to train on, it would unnecessarily expand the output layer. Having classes in the output layer without corresponding samples can lead to the model learning to ignore these classes, which skews the training process.

To address this, we create a class mapping that converts the original class labels to a new class index range consisting only of labeled classes (N_L). This class index range is also the possible output of the model, the output layer. Classes that are not yet labeled (N_U) are mapped to -1 and do not exist in the output layer. When novel classes are discovered (as a result of annotating, N_+) from the unlabeled datasets, the class index range is extended, incorporating the new class into the output layer. This process is illustrated in Figure 4. Samples in D_{open} are also labeled as class -1, allowing for OSR evaluations to be conducted on classes mapped to -1.

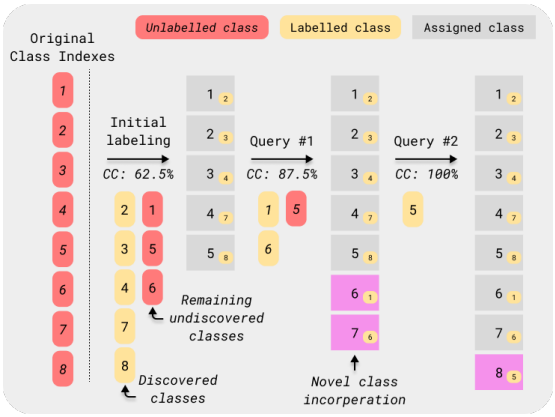

Fig. 4: Illustration of the class reorder process. Only the classes in D_L are represented in the output layer. We assign an alternative class index to each discovered class so that the output layer holds only labeled classes. If a novel class is discovered during the annotation process, the output layer is extended to include the new class. *CC* denotes Class Coverage, which is the percentage of classes discovered in the dataset.

This simple yet efficient method allows the model to dynamically incorporate new classes without needing to know the number of classes in the dataset beforehand. It also enables the model to continuously train on an increasing number of trainable classes without retraining the entire model (as elaborated in Section III-D1), addressing a common issue in CIL and OSR methods.

4) K-Fold cross-validation: During CIL cycles, we use K-Fold cross-validation to create D_{train} and D_{val}. K-Fold cross-validation ensures that the model is evaluated on different subsets of the data, providing a more robust estimate of model performance and reducing the risk of overfitting to a static D_{val}. A static D_{val} is impractical in this setting, as new data is labeled continuously, requiring a new D_{val} to be created for every epoch.

The dataset is split into K folds, where each fold serves as the validation set D_{val} once. The model is trained on D_{train}, which consists of the remaining $K - 1$ folds. Performance evaluation is conducted on the generated D_{val}, where the same early stopping mechanism is applied. No class balancing is performed on D_{val} as it best simulates the distribution of D_U. Increasing K reduces the amount of data in the validation set but maximizes the use of data for training.

The model is trained with an early stopping mechanism based on validation loss on D_{val}. If no improvement is observed for a set number of epochs (Patience), the model reverts to the state with the highest validation accuracy, and training is stopped.

5) Balanced sampling: To mitigate the effects of class-imbalanced learning, where head classes are overrepresented compared to tail classes, we use a stratified sampling method, Random Oversampling (ROS), to balance the number of samples presented per class. For each epoch, a new variation of D_{train} is created, where each represented class gets a configurable number of slots. Samples are randomly selected from the available samples for that class (see Section VII-D). If a class has fewer samples than the number of slots, already chosen samples can be picked again until the number of slots is filled. Data augmentation is applied separately to these duplicate samples, generating different variations of the same data sample.

D. Pretrained Model Weights

In our research, we approach the use of pretrained models in two distinct ways: using open-source available datasets (and weights) and learning feature representations from an unlabeled dataset.

a) Pretrained model on ImageNet: As a general open-source pretrained model, we utilize the provided weights from PyTorch's [57] for a ResNet50 [58] network pretrained on the full ImageNet dataset [59]. Our evaluations on ImageNet-LT using these weights should be interpreted with caution, as the ResNet50 backbone has been trained on the same dataset, which could affect the evaluation of its ability to discover novel classes. We argue that evaluating on ImageNet-LT extends this research by providing a case where an almost "perfect" pretrained or transferable model exists.

b) Unsupervised Feature Representation Learning: As an alternative to using a general pretrained model, we explore learning feature representations on an unlabeled dataset. This approach is motivated by the idea that the model can learn domain-specific features that may not be present in a general pretrained model. Furthermore, by using Unsupervised Feature Representation Learning rather than a supervised approach, our research supports applications where open-source pretrained models for transfer learning may not be available. We use an unsupervised learning approach called MoCo [52, 53], where the model is trained through contrastive learning on an unlabeled dataset. It is important to note that the resulting features focus on distinguishing characteristics, which may be more suitable for our task of discovering novel classes.

We train our MoCo models prior to our experiments using 4 distributed GPUs with a batch size of 256, following the suggested hyperparameters in the original paper. For each dataset, a model is trained for 600 epochs using only the images (not any labels). The resulting performance is shown in Table A.3.

1) Class-Incremental Learning: After D_I is created, the CIL process begins. In CIL, the model iteratively queries samples from D_U and adds them to D_L, incorporating novel classes. K-Fold cross-validation and balanced sampling together form part of CIL's Replay-based tasks. In replay-based tasks, the network revisits former classes by replaying samples from previous tasks. This mitigates catastrophic forgetting, where the introduction of new classes can cause the model to forget previously learned ones. Unlike traditional CIL tasks, which are predetermined at the start of training, our tasks are generated at each epoch. An epoch is defined as a single pass of all samples provided by the balanced sampler, and an iteration refers to each individual cycle of training and annotating samples. Each class is given a configurable number of slots, meaning novel class samples may be presented more than once during training. Classes with more samples are limited to presenting a subset of their samples per epoch, allowing the model to revisit former classes similarly to replay-based tasks.

2) Training Configurations: We avoid using similarity-based networks, which are common in FSCIL tasks, where the model is trained to recognize known classes and detect unknown classes based on their similarity to the known classes. Similarity-based networks do not scale well to large datasets [50, 51], as they require changes in hierarchical class structures. These changes introduce additional complexity and instability, which are undesirable in our setting because they make it harder to maintain consistent performance and scalability.

Instead, we use traditional class-classification networks and cross-entropy loss. Class-classification networks categorize input data into predefined classes, and cross-entropy loss measures the difference between the predicted probability distribution and the true distribution, effectively penalizing incorrect classifications.

The model is trained using a single Adam optimizer split over the backbone and the Fully Connected layer (FC) layer:

- **Backbone**: The backbone is trained using a *OneCycleLR* scheduler, which starts with a learning rate of 0, increases to 1.5×10^{-4}, and then decreases back to 0.
- **FC**: The FC is trained using a *StepLR* scheduler, which reduces the learning rate by a factor of 0.9 every 5 epochs.

These schedulers allow the model to first train the FC layer to learn novel classes and then fine-tune the backbone to incorporate these new classes. Without this implementation, the randomly initialized FC neurons for novel classes could interfere with the backbone's training, hindering the fine-tuning process.

E. Evaluation Metrics

During this research, we have access to the ground truth of the dataset, which allows us to evaluate the detectors directly on N_U. To avoid biases related to threshold selection, we evaluate the detectors in a threshold-independent manner, in line with our approach of selecting a fixed number of samples to label in each iteration, as detailed in Section III-C2. Our evaluation focuses on Area Under Receiver Operating Characteristic (AUROC) and Area Under the Precision-Recall (AUPR), common metrics in the field of OSR, which are based on single-step evaluations. For our CIL experiments, we incorporate additional metrics: Queried Novel Classes, Class Coverage, and Effort metrics, which assess the overall efficiency and effectiveness of the entire annotation process beyond a single OSR step.

We utilize the following metrics to evaluate the detectors:

- **AUROC**↑: The Area Under Receiver Operating Characteristic measures the detector's ability to distinguish between known and novel classes across all possible thresholds. It represents the probability that a randomly chosen novel class will be ranked higher than a randomly chosen known class. AUROC is useful for balanced datasets where both classes are equally important, but it can be less informative in imbalanced scenarios, as it may give an overly optimistic view of performance.
- **AUPR**↑: The Area Under the Precision-Recall [60] quantifies the area under the precision-recall curve, capturing the trade-off between precision and recall across all thresholds. AUPR is especially valuable in imbalanced datasets, such as our setting, where the positive class (novel classes) is rare, making precision as important as recall. This metric provides a more focused evaluation of the detector's performance in such scenarios.
- **Queried Novel Classes** (N_+)↑ **and Novel samples** (x_+)↑: By sampling D_+ samples from D_U and querying the oracle, D_+ may contain N_U. This metric provides insight into the model's ability to discover new classes during a single annotation step. For D_U, we use N_+ to denote the discovered novel classes, where each novel class is represented by a newly labeled sample. For D_{open}, we use x_+ to denote the discovered novel samples, as these datasets do not contain class labels, and therefore N_+ cannot be determined.
- **Class Coverage**↑: This metric represents the percentage of classes discovered in the dataset and is calculated as the percentage of classes with at least one labeled sample in D_L (Class Coverage $= |N_L|/|N|$). It provides insight into the model's ability to discover new classes over the course of the entire annotation process.
- **Effort** (D_L/D)↓: This metric tracks the total number of labeled samples in D_L compared to the total number of samples in D. Used in conjunction with Class Coverage, it evaluates the efficiency of the model in discovering new classes while minimizing the human labeling effort.

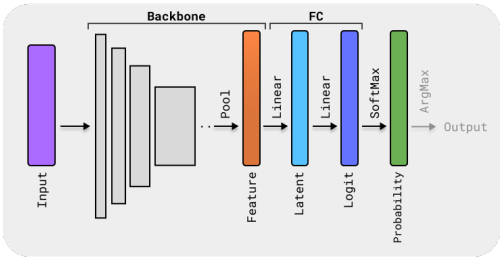

Fig. 5: Illustration of the detector categories defined in this research. Each category corresponds to a specific state of the data being processed by the model. The backbone consists of the convolutional layers, the FC is the fully connected layer, and the Softmax and ArgMax operations result in the final classification of input data.

F. Novel Class Detection

In line with AL terminology, our sampling strategies select the most informative samples from D_U to be queried by the oracle. These strategies, referred to as "detectors," are used to infer over the entire D_U, assigning a novelty score to each unlabeled sample. The $|D_+|$ samples with the most favorable novelty scores are selected, labeled by the oracle, and added to D_L.

We categorize the detectors based on which data layer they use as input for their algorithm. Figure 5 illustrates the different categories with respect to a model's architecture. The categories are as follows:

- **Probability**: Uses the softmax class probabilities as input.
- **Logit**: Operates on the output of the FC layer before the softmax layer.
- **Latent**: Uses the linear layer between the backbone and the FC layer.
- **Feature**: Takes the output from the backbone, after the last pooling layer (except for ASH detectors).
- **Input**: Utilizes the entire model to process input features.

Table I lists all detectors used in this research, providing a short description and their corresponding category. The detectors are implemented based on the framework by [61].

IV. Experiment Results

We conducted a series of experiments aimed at investigating the distinctions between traditional training paradigms and our novelty detection task, as well as evaluating how different novelty detection approaches perform in discovering novel classes during the annotation process. Since the detectors employ distinct methodologies for novelty detection, we systematically tested multiple configurations for each. These configurations explored variations in pretrained models (using ImageNet and MoCo) and trainable parameters. Each experiment was repeated twice with different random seeds to ensure robustness, and the results were averaged. For clarity, we present only the best-performing configuration for each detector, where "best" refers to the configuration that achieves the highest average ranking across all points on the x-axis in the respective graphs.

DETECTOR	USES	DESCRIPTION	SOURCE
Uncertainty	Probability	Uses softmax output probabilities to compute the uncertainty score of a sample, identifying how likely the sample belongs to the predicted class.	Ours
Margin	Probability	Calculates the margin score by taking the difference between the highest and the second-highest class probabilities.	Ours
KLMatching	Probability	Captures the typical posterior distribution shape for each class and compares the network's softmax distribution during inferencing to these templates, generating an anomaly score based on the minimum KL divergence.	[30]
Energy	Logit	Computes the negative energy of the logit vector, which is utilized as an outlier detection score to identify novel classes.	[31]
Entropy	Logit	Measures the entropy of the classifier's logits to quantify uncertainty in the model's predictions, aiding in detecting outliers.	[32]
KNN	Latent	Fits a k-Nearest Neighbor model to labeled samples and scores unlabeled samples by calculating their distances to the nearest labeled neighbors, using these distances as an outlier score. *(Minkowski Euclidean distance, k:3, radius:1, leaf size:30, scoring:Energy)*	[34]
ViM	Logit+Feature	Detects OOD samples by generating a virtual logit from the residuals in the feature space and matching it with the original logits. The softmax probability of this virtual logit indicates the degree of OOD-ness. *(Dimensionality of the principal subspace:0)*	[40]
ReAct	Feature	Identifies the most influential weights post-backbone layer using a contribution matrix, then sparsifies the connections based on these weights, improving OOD detection without modifying the network's parameters. *(Clipping threshold:1, scoring:Energy)*	[33]
ASH	Feature	Prunes the largest activations from the features (ASH-p), binarizes the remaining activations (ASH-b), and rescales them (ASH-s), with energy-based outlier scoring. *(Percentile activations modified:0.65, scoring:Energy)*	[35]
DICE	Feature	Sparsifies weights by ranking them according to their contribution to ID classification and selects the most significant ones for detecting OOD. *(Percentile weight drop:0.7, scoring:Energy)*	[36]
SHE	Feature	Uses a Hopfield Network to store patterns and retrieve them by minimizing an energy function. This process updates input patterns to converge toward stored patterns. OOD detection is facilitated by comparing input patterns with those derived from the labeled dataset, measuring their similarity.	[37]
RMD	Input+Feature	Enhances the Mahalanobis Distance (MD) method by subtracting the MD of a distribution fitted on all training data from the class-specific MD, effectively computing a likelihood ratio. This provides a robust, hyperparameter-free confidence score for near-OOD detection.	[39]
ODIN	Input	Enhances the separation of ID and OOD data by adjusting softmax scores with temperature scaling. It then preprocesses inputs by adding perturbations that increase the softmax score, enabling effective classification based on a predetermined threshold. *(Gradient descent step:0.05)*	[38]
Committee	-	Combines multiple detectors, elaborated in Section IV-D.	Ours

TABLE I: Overview of novelty detectors used in this research, including their hyperparameters (if applicable).

The first experiment (Section IV-A) seeks to understand the complexity of novelty detection in our specific setting, questioning whether performance on the OSR task directly translates to performance in our task. We assess the detectors' performance at a single step of the annotation process using the D_U and D_{open} datasets. To explore the influence of different training settings, we introduce the following:

- **Both**: Both the backbone and the Fully Connected layer (FC) layer are trained.
- **Frozen**: Only the FC layer is trained, while the backbone remains frozen.
- **Pretrained model**: The pretrained model is either supervised (ImageNet) or unsupervised (MoCo).

We evaluate performance using AUROC and AUPR, which are standard metrics in OOD and OSR evaluations. Additionally, we introduce custom metrics, x_+ and N_+, to measure annotation effectiveness in our task. Early stopping is applied based on a predefined validation set to mitigate overfitting.

In the second experiment (Section IV-B), we investigate whether employing K-Fold cross-validation, as opposed to using the predefined validation set, affects overfitting. By splitting the validation data from the training set, we introduce additional complexity into the training process and analyze its effect on model generalizability.

The third experiment (Section IV-C) investigates how different detectors perform in discovering novel classes during the annotation process. Here, we compare detectors and random sampling in terms of achieving Class Coverage, as indicated by the number of labeled samples required. Upon identifying novel classes (N_+), the model's output layer is extended to accommodate these new classes (see Section III-C3). Two different approaches are tested:

- **Continuous**: Both the backbone and FC layer are kept as new classes are discovered. The additional outputs in the FC layer are randomly initialized along with the optimizer. This approach raises concerns about potential overfitting, which we further elaborate on in Section IV-B.
- **Reload**: The backbone is reset to its original pre-trained state, and the FC layer and optimizer are reinitialized. This approach completely restarts training with newly discovered classes, providing insights into potential improvements in class discoverability.

The final experiment (Section IV-D) examines the potential for leveraging multiple detectors simultaneously. We investigate whether employing multiple detectors in a committee approach can enhance novelty detection performance and assess whether a universal combination of detectors and configurations can perform consistently across diverse datasets.

A. Single-Step Novelty Detection and Open-set Recognition

In the first experiment, we evaluate detector performance in a single-step annotation process using the D_U and D_{open} datasets. The evaluation metrics include Area Under Receiver Operating Characteristic (AUROC), Area Under the Precision-Recall (AUPR), x_+, and N_+, with AUROC and AUPR being standard in Out-of-Distribution (OOD) and Open-Set Recognition (OSR) evaluation. The x_+ and N_+ metrics emphasize practical performance by selecting the top-scoring $|D_+|$ samples (following the configuration in Table A.4). In this experiment, we use the dataset's **Predefined Validation set** for early stopping, ensuring that the model does not overfit to the training data.

For D_{open}, which reflects traditional OOD and OSR tasks, AUPR benefits from having an increased amount of labeled data. As shown in Figure 6, detectors maintain relatively stable AUROC values across all labeled percentages, reflecting consistent binary classification ability as the class distribution shifts. In contrast, D_U presents a more challenging task, requiring the discovery of novel classes within the same dataset. As more data is labeled, the AUPR and N_+ metrics exhibit a downward trend due to the decreasing number of novel classes in D_U, making novel instance detection increasingly difficult.

The ranking of the detectors would be expected to be similar on both D_U and D_{open} if the settings were comparable. However, this is not the case for AUROC, AUPR, x_+, and N_+. When analyzing the overall ranking of the detectors across all datasets, we observe that the best and worst-performing detectors alternate between D_U and D_{open}. This finding suggests that detector performance on D_{open} does not reliably generalize to D_U, indicating that the two tasks differ due to variations in dataset structure and complexity.

A noteworthy finding is that for several detectors, the unsupervised MoCo pretrained model performs better on the D_{open} datasets than the ImageNet pretrained models, though this advantage does not extend to the D_U datasets, contrary to our expectations. Given the unique features learned through contrastive learning in MoCo, we anticipated better performance on D_U. This suggests that MoCo's learned features may be more effective in OSR tasks than those learned through traditional supervised classification. Notably, even when compared to the almost perfectly pretrained ImageNet models on ImageNet-LT, MoCo achieves superior performance on the D_{open} dataset for several detectors, further highlighting its potential to enhance OSR capabilities.

In summary, this experiment highlights the differences between traditional OOD and OSR tasks and our novelty detection task. It demonstrates the complexity of novelty detection during the annotation process and the limitations of traditional metrics and methods in this setting. Each detector has an optimal configuration for each dataset and metric. Surprisingly, the MoCo pretrained model does not show significant benefits for our novelty detection tasks, although it does for OSR tasks. No further analysis is provided for the OSR tasks, as the focus of this research is on novelty detection. Overall, the results suggest that the best detectors for our Active Class-Incremental Learning task are KLMatching, RMD, DICE, Uncertainty, KNN, Margin, and SHE. However, their performance may depend on the dataset, the amount of labeled data, and the multi-step nature of the problem, which we further explored in Section IV-C.

B. Training using K-Fold Cross-Validation

In this experiment, we no longer use the predefined validation dataset for early stopping, as in the first experiment (Section IV-A). Instead, the validation data is split from the training data, aligning with our setting where preventing overfitting is critical. We present the results for Places365-LT, noting that ImageNet-LT shows similar behavior (see Figures A.9 and A.10) due to shared characteristics (e.g., long-tailed (LT) distribution, annotation quality, and class imbalance). The key differences in performance are primarily driven by the training paradigm rather than the dataset itself. iNaturalist2018-Plantae is less affected by K-Fold cross-validation, likely due to a too small model size to train effectively.

1) Accuracy Performance and Trainable Settings: We utilize accuracy as a metric to evaluate the training performance and to assess the impact of K-Fold cross-validation on overfitting and generalization. In Figure 7, we compare the training performance for each combination of pretrained models and training settings. The results show that the **MoCo** pretrained model has notably lower training and testing accuracy compared to the **ImageNet** pretrained model. This is likely due to **MoCo** being trained with a contrastive loss function, which emphasizes the extraction of more unique features for class distinction [54], but tends to underperform in classification tasks. In the first experiment (Section IV-A), we demonstrated that the MoCo pretrained model performs better on the D_{open} dataset with several detectors in the OSR task, suggesting that testing accuracy does not always correlate with performance in novelty detection tasks.

2) K-Fold Cross-Validation: Compared to using a predefined validation set, K-Fold cross-validation introduces a higher risk of overfitting, as the model is evaluated on the same data it was trained on (see Section III-C4). The more the model is trained, the less effective the validation process becomes, as demonstrated in the **Both Continuous** setting. Overfitting is typically indicated by high training accuracy and lower testing accuracy, which is evident in Figure 7. When both the backbone and FC layer are trained, the training accuracy is higher, but the testing accuracy is lower compared to freezing the pretrained model. In contrast, in the **Frozen** setting, the

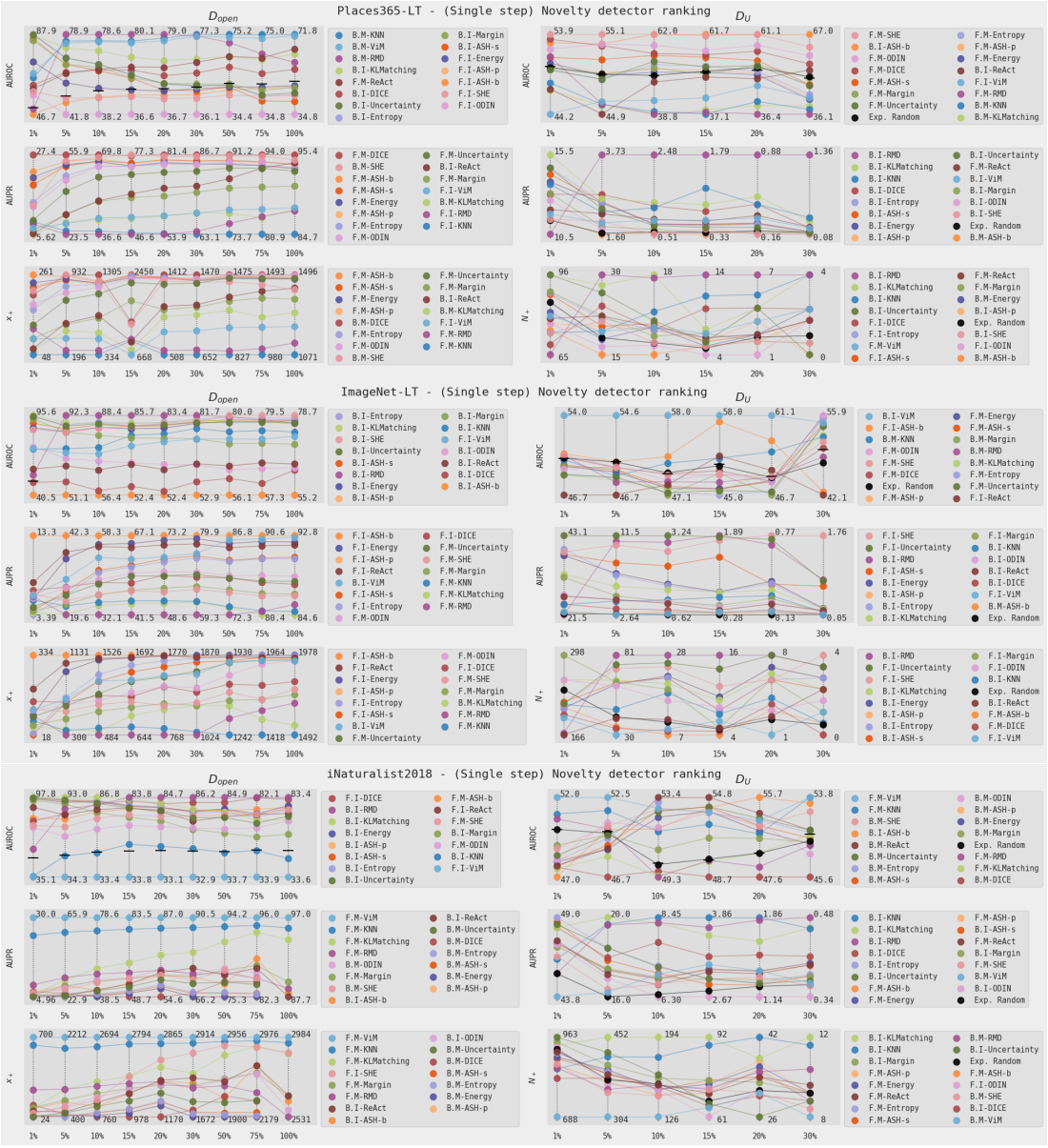

Fig. 6: Open-set recognition on the Places365-LT and ImageNet-LT datasets using the predefined validation set. The detectors are evaluated on the D_{open} dataset (left) and the D_U dataset (right) with the metrics AUROC↑ (top), AUPR↑ (middle), and x_+ and N_+ ↑ (bottom). The x-axis indicates the percentage of the dataset that is labeled D_L/D ↓, and the y-axis is scaled to the range of the minimum and maximum values for each labeled percentage. We include *Experimental Random*, which shows values obtained using random sampling. In AUROC plots, the horizontal line indicates a AUROC of 0.5. Only the best configuration per detector is shown; the legend indicates the configuration (e.g., **B.I** implies **B**oth on **I**mageNet's pretrained model, and **F.M** implies **F**C on **M**oCo's pretrained model). The legend is sorted based on the average ranking of detectors across all labeled dataset percentages.

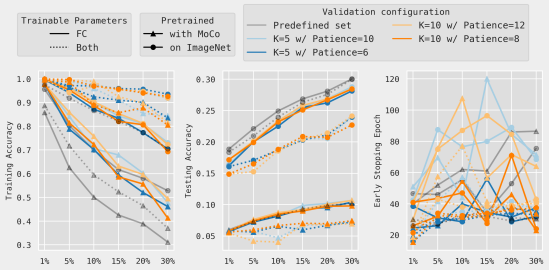

Fig. 7: Impact of K-Fold cross-validation on training and testing accuracies using different training settings and pretrained models on Places365-LT. The x-axis indicates the percentage of the dataset that is labeled. Training and Testing Accuracies↑ are shown on the y-axis. The rightmost graph shows the average Early Stopping Epoch↓ for each k-fold configuration.

testing accuracy is less affected by K-Fold cross-validation, indicating that training both components increases the risk of overfitting.

In Figure A.9, we analyze the impact of K-Fold cross-validation on the single-step novelty detection task. Despite the reduced generalization, most detectors show improved performance when training both components, likely due to the detectors' enhanced ability to distinguish between novel classes. Therefore, all configurations (training settings and pretrained models) are still considered.

Both Figure Figure 7 and Table A.9 display different configurations of K-Fold cross-validation, where the number of folds and patience values are varied. The results indicate a minimal impact on accuracies and detection performance, except for the Early Stopping Epoch, which is significantly affected. A **K of 5 with a Patience of 6** produces the lowest Early Stopping Epoch across all combinations of training settings and pretrained models and will be applied in the subsequent experiments.

When the backbone is not **Frozen**, K-Fold cross-validation exacerbates overfitting, especially in the **Continuous** setting, where the backbone is kept throughout the annotation process. Over successive iterations, the validation data is drawn from the training data that has already been fine-tuned, leading to an ineffective validation set. The effects of this issue are further explored in Section IV-C.

C. Novelty Detection with Active Class-Incremental Learning

Using the detectors from Table I, we apply the Class-Incremental Learning (CIL) framework (see Figure 1). We evaluate detector performance based on the number of labeled samples D_L required to achieve Class Coverage. After selecting the initial dataset D_I at random, we follow the cycle described in Section III-D1. The model is trained on D_L, novelty detection is used to select $|D_+|$ samples from D_U for labeling, these samples are added to D_L, and novel classes are incorporated into the model through CIL. This process is repeated for 14 more queries, resulting in 15 queries including D_I. During this process, we record class coverage, paying special attention to the number of samples required to achieve 90%, 95%, 99%, and 100% class coverage (milestones). The experimental configurations per dataset are shown in Table A.4, which also includes the milestones for a naive setting where samples are selected randomly. We vary the following parameters: the detector, training settings (**Frozen**, **Both**), model state retention (**Continuous**, **Reload**), and the pretrained model (**ImageNet**, **MoCo**). Each experiment is conducted twice with seeds 1 and 2, and the results shown are averaged across both seeds. Early stopping is performed on D_{val} using **K-Fold cross-validation**, with a K of 5 and a patience of 6.

The best configurations for each detector are presented in Figure 8, with a comprehensive overview provided in Table A.6. As expected, the N_+ metric in single-step detection (Section IV-A and Section IV-B) serves as a good indicator of performance in our multi-step novelty detection task.

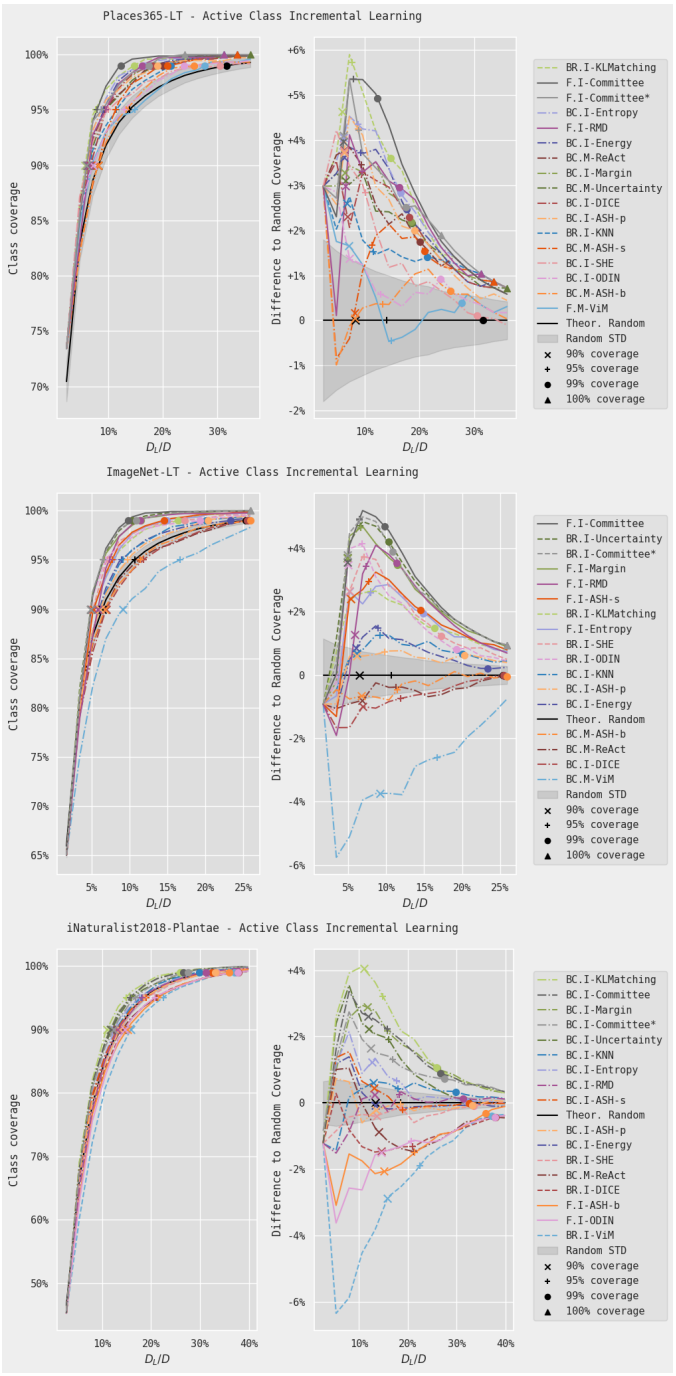

Fig. 8: Best-performing detectors across all datasets in the Novelty Detection with Active Class-Incremental Learning process. The x-axis indicates the percentage of the dataset that is labeled $D_L/D \downarrow$, and the y-axis shows corresponding Class Coverage. Milestone markers indicate when the detector achieves Class Coverage. The mean and standard deviation for random sampling are included. The right plot shows the same data, with the y-axis representing the difference from the theoretical Random baseline. The legend is sorted by the average ranking of detectors across all queries.

However, due to the sequential nature of multi-step detection, a critical nuance is overlooked in single-step experiments: after each step, the samples queried by the detector are removed, making subsequent detections progressively more challenging. Additionally, some detectors may develop a bias toward certain features or classes, reducing their ability to discover novel classes in later steps. Notably, most detectors from the **Feature** category exhibit this behavior, where they rank better in the single-step experiments than in the multi-step ones. The results also demonstrate that the iNaturalist2018-Plantae dataset presents significant difficulty in identifying all classes, which can be attributed to a more imbalanced class distribution, as shown in Figure 3.

Detectors from the **Probability** category, as well as RMD, demonstrate superior performance in multi-step novelty detection. As shown in Figure 8, Table A.6, and Table A.5, these detectors consistently rank among the top performers across all datasets. Table A.6 details the number of labels required to reach specific milestones compared to random sampling, highlighting the best individual detectors for each dataset. By using KLMatching, RMD, Entropy, and KNN on Places365-LT, we achieve *100%* class discovery with **46.9%** fewer annotations compared to random sampling. Similarly, on ImageNet-LT and iNaturalist2018-Plantae, we achieve *99%* class coverage with **59.9%** fewer annotations using Uncertainty and **23.0%** fewer annotations using KLMatching, respectively, all leveraging the supervised **ImageNet** pretrained model.

D. Multiple Detectors by Committee

In this final experiment, we explore the benefits of using multiple detectors to discover novel classes during the annotation process. In the previous experiment (Section IV-C), we observed significant variation in detector performance across datasets. Some detectors excelled on specific datasets, and their performance was influenced by the amount of labeled data available. As each queried novel class is removed from the pool of undiscovered classes, subsequent detection becomes more challenging, and a detector's novelty detection potential may diminish over time. We investigate the potential benefits of combining detectors [62] to perform consistently well across various datasets, regardless of their characteristics (e.g., label semantics).

1) Potential Novel Classes Using Other Detectors: We assessed the benefits of multiple detectors by analyzing how many novel classes each alternative detector would have queried at a single step, had it been used instead of the active detector. The results, visualized in Table A.5, confirm that different detectors identify different novel classes. This supports the hypothesis that combining detectors can enhance overall performance.

2) Combining Detectors: Based on the results from Table A.5, we selected the best-performing detectors for each pre-trained model and trainable parameter. The combinations are shown in Table II.

PRE-TRAINED	DETECTOR TRAIN. PAR.	RMD	KLMatching	Margin	Uncertainty	ReAct	KNN	Entropy
ImageNet	Both Cont.	x	x	x	x		x	x
	Both Reload	x	x	x	x			
	Frozen	x		x	x			
MoCo	Both Cont.		x		x	x	x	x
	Both Reload	x	x	x		x		
	Frozen	x	x	x		x		
	T-Scaling	$\min(0+0.1x,\ 1.1)$	$\max(1.5-0.1x,\ 1.1)$	$\max(1.2-0.1x,\ 1.0)$	$\max(1.5-0.1x,\ 1.1)$	$\max(1.1-0.05x,\ 1.0)$	$\min(0.8+0.05,\ 1.0)$	0.8

TABLE II: Multiple detector combinations used for each pre-trained model and trainable parameter. The T-Scaling column shows the scaling factor used for each detector, where x represents the query number.

We propose two approaches for combining detectors:

- **Committee by Majority (Committee):** Each detector selects its top-ranked novelty samples, with the weight of each vote based on the sample's rank in the novelty score. The top $|D_+|$ samples with the highest collective weight are queried. Samples selected by multiple detectors are more likely to belong to novel classes and are prioritized for querying.
- **T-scaled Committee by Majority (Committee*):** This method extends the Committee by Majority approach by applying a temperature scaling factor, adjusting detector weights based on their performance. This accounts for the observation that some detectors perform better in the early stages of annotation, while others excel later in the process. The scaling factors used in this experiment are shown in Table II.

The comparative performance of these combinations against individual detectors is presented in Figure 8 and Table A.6, labeled as Committee and Committee*, ranking among the highest-performing detectors. The T-scaled Committee* approach consistently outperforms individual detectors, achieving *100%* class coverage for Places365-LT and ImageNet-LT with **59.1%** and **57.6%** fewer annotations than random sampling, respectively.

On iNaturalist2018-Plantae, however, it was still outperformed by Margin and KLMatching. We anticipate that further fine-tuning of the T-scaling factors will enable Committee* to outperform on iNaturalist2018-Plantae as well. These results underscore the potential of multi-detector approaches to significantly enhance annotation efficiency across datasets.

V. Discussion

The experiments conducted in this study offer several significant insights that advance the field of novelty detection and long-tailed (LT) dataset annotation. First, our results highlight a clear distinction between the Open-Set Recognition (OSR) task and the novelty detection tasks introduced in this study. This distinction implies that the evaluation metrics and methodologies used in existing OSR and Out-of-Distribution (OOD) research are not directly transferable to our setting (see Section IV-A). This realization underscores the necessity for a dedicated benchmark specific to novelty detection, which our study establishes by providing a thorough evaluation of multiple detection algorithms across three widely used LT datasets with various training settings.

Furthermore, the integration of OOD detection methods, specifically KLMatching, RMD, and ReAct, into Active Learning (AL) sampling strategies demonstrated a notable improvement in discovery performance, both for OSR and in our novelty detection task. These detectors, combined with traditional AL strategies such as Uncertainty, Margin, KNN, and Entropy, consistently ranked among the top performers across all datasets. This observation suggests that combining AL with OOD detection methods enhances the identification of valuable samples for annotation, which is the primary aim of AL. The combination of multiple detectors through the Committee approach consistently ranked among the top performers across all datasets, leveraging the complementary strengths of various detectors to provide a more robust solution for novelty detection.

Another key finding of this study is the exploration of the usability of unsupervised pre-trained models, such as MoCo, illustrating their potential in scenarios where supervised models are unavailable. Although these models generally underperformed compared to supervised pre-trained models (on ImageNet), they achieved a significant reduction in annotations needed to achieve high class coverage, demonstrating that unsupervised learning is a viable alternative in contexts where transfer learning from supervised models is not feasible.

Lastly, specific detectors have preferences for training settings; however, it is possible to train a model where detectors can discover novel classes while the backbone is frozen with any non-feature-based detectors. This eliminates the need for class-incremental learning techniques used during continuous training, simplifying and accelerating the training process. Selecting the optimal training setting, detector, or combination, and pre-trained model depends on the dataset and resource availability. However, our results demonstrate that it is feasible to discover all classes with significantly fewer annotations than random sampling across all datasets using probability-based detectors while the pre-trained backbone is frozen.

VI. Conclusion

This research presents a novel approach at the intersection of AL, Class-Incremental Learning (CIL), and novelty detection, with a particular focus on annotating long-tailed image datasets. Our method operates under an open-set assumption and facilitates continuous learning as novel classes emerge, without relying on predefined validation sets or prior class knowledge.

Our findings reveal that the detectors KLMatching, RMD, and ReAct from the field of OOD detection improve on the standard AL strategies, such as Uncertainty, Margin, KNN, and Entropy, in identifying valuable samples for annotation. KLMatching discovers *99%* of the classes with 23.0% fewer annotations than random sampling on the iNaturalist2018-Plantae dataset. The Committee approach, which combines multiple detectors, consistently ranked among the top three across all datasets, further validating its potential. Specifically, it discovers *100%* of the classes on Places365-LT and ImageNet-LT with 59.1% and 57.6% fewer annotations, respectively.

In conclusion, our research provides novel insights and establishes a foundation for future research in the field, particularly at the intersection of Active Learning, Class-Incremental Learning, and novelty detection, with clear implications for real-world use cases. This work provides a scalable and cost-efficient methodology for annotating long-tailed datasets under an open-set assumption, bridging the gap between theoretical novelty detection research and practical applications.

VII. Future Work

This study aimed to establish a foundation for future research rather than provide a fully optimized solution. Many hyperparameters, including detector-specific settings (see Section I), training hyperparameters, and the T-scaling factors for the Committee* approach (see Table II), were not extensively fine-tuned. Further exploration of these parameters could result in performance improvements, and running the experiments on more seeds will provide a more accurate evaluation. Nevertheless, the provided evaluation demonstrates the effectiveness of the proposed methodology in reducing annotation effort and improving class discovery in long-tailed datasets. Several promising directions for future research, based on the insights gained from this study, are discussed below.

A. Impact of Label Semantics

One underexplored aspect is the role of label semantics in the performance of novelty detection. Our use of three distinct types of label semantics (see Section III-B2) suggests that semantics influence detector performance, specifically on MoCo-pretrained models. These models, trained uniformly across datasets, still exhibited preferences for specific datasets, supporting the hypothesis that label semantics affect discovery performance. Future research should investigate this further,

ideally with additional datasets having comparable label semantics to validate these findings.

B. Unsupervised Learning

This study reveals considerable room for exploring the potential of unsupervised learning in long-tailed datasets. Several avenues for future work include:

- **Effect of Long-tailedness on Unsupervised Learning:** Investigating how long-tailed distributions affect the performance of unsupervised representation learning could provide valuable insights.
- **Alternative Pre-training Methods:** Exploring methods like SimCLR [63] or BYOL [64] may enhance the performance of unsupervised models. The work of [65] could serve as a starting point.
- **Unsupervised Classification for D_I:** Instead of random initialization, unsupervised classification techniques could be used to select the most informative samples for annotation, improving class coverage and label diversity at the start of the process.

C. Exploration of Alternative Architectures

The selection of ResNet50 as the backbone for detectors balanced performance with computational efficiency, as well as its widespread use in research. However, exploring different architectures may reveal improvements in performance or resource use. Future research could explore:

- **Model size:** Examining smaller models like ResNet18 or wider variants like WideResNet may yield insights into the trade-offs between performance and computational costs, as well as the impact of labeled dataset size on model complexity.
- **Vision Transformers:** Vision transformers have demonstrated superior performance in image classification, particularly when capturing global context [66]. Integrating transformers with existing detector algorithms, especially unsupervised methods like MoCo [53], could be a promising direction.
- **Prototypical Networks:** Extending the dataset structure to incorporate hierarchical elements and employing Prototypical Networks, as used in few-shot learning Few-Shot Class-Incremental Learning (FSCIL), may improve detection performance.
- **Model Ensembling:** Combining models of varying architectures, pre-training methods, or training settings into an ensemble could improve detection results, as seen with the Committee approach (see Section IV-D).
- **Liquid Neural Networks:** Adopting Liquid Neural Networks [67], which dynamically adjust model size based on dataset complexity, may allow models to start small and scale up as needed with dataset complexity.

D. Optimization of Sample Selection Techniques

We employed Random Oversampling (ROS) to mitigate class imbalance (see Section III-C5). However, this method does not consider sample informativeness. Future work could refine sample selection by incorporating the novelty detection scores, which already emphasize a sample's uniqueness. This prioritization could allow the model to better generalize for novel classes. Additionally, leveraging Explainable AI (XAI) techniques such as SHAP [68] or LIME [69] could help identify influential samples, guiding more informed resampling strategies.

E. Post-process Accuracy Optimization

This study concludes with a (near) closed-set dataset. Further research could investigate how testing accuracy compares when using the obtained dataset with fewer annotations versus the entire dataset, providing insights that enable more meaningful comparisons with related work on AL.

References

[1] Grant Van Horn, Oisin Mac Aodha, Yang Song, et al. The inaturalist species classification and detection dataset. *arXiv*, 2017. doi: 10.48550/arxiv.1707.06642.

[2] Zhixiong Yang, Junwen Pan, Yanzhan Yang, et al. Proco: Prototype-aware contrastive learning for long-tailed medical image classification. 2022.

[3] Ruru Zhang, Haihong E, Lifei Yuan, et al. Mbnm: Multi-branch network based on memory features for long-tailed medical image recognition. *Computer Methods and Programs in Biomedicine*, 212:106448, 2021. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2021.106448. URL https://www.sciencedirect.com/science/article/pii/S0169260721005228.

[4] Zhenzhen Weng, Mehmet Giray Ogut, Shai Limonchik, et al. Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. *arXiv*, 2021. doi: 10.48550/arxiv.2104.01257.

[5] Burr Settles. Active learning literature survey. 2009. URL https://api.semanticscholar.org/CorpusID:324600.

[6] Dan Pelleg and Andrew Moore. Active learning for anomaly and rare-category detection. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://proceedings.neurips.cc/paper_files/paper/2004/file/8c59fd6fbe0e9793ec2b27971221cace-Paper.pdf.

[7] Jingrui He and Jaime Carbonell. Nearest-neighbor-based active learning for rare category detection. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/2838023a778dfaecdc212708f721b788-Paper.pdf.

[8] 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.

[9] Tom Haines and Tao Xiang. Active rare class discovery and classification using dirichlet processes. *International*

Journal of Computer Vision, 106, 02 2014. doi: 10.1007/s11263-013-0630-3.

[10] Yuzhen Chen and Haibo Ye. Improving active learning on imbalanced datasets by features mixing. *International Conference on Cloud Computing, Performance Computing, and Deep Learning (CCPCDL 2023)*, 12712: 1271218–1271218–10, 2023. ISSN 0277-786X. doi: 10.1117/12.2678956.

[11] Chaozheng Wang, Shuzheng Gao, Cuiyun Gao, et al. Label-aware distribution calibration for long-tailed classification. *arXiv*, 2021. doi: 10.48550/arxiv.2111.04901.

[12] Chen Wei, Kihyuk Sohn, Clayton Mellina, et al. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. *arXiv*, 2021. doi: 10.48550/arxiv.2102.09559.

[13] Qiuye Jin, Mingzhi Yuan, Haoran Wang, et al. Deep active learning models for imbalanced image classification. *Knowledge-Based Systems*, 257:109817, 2022. ISSN 0950-7051. doi: 10.1016/j.knosys.2022.109817.

[14] Javad Zolfaghari Bengar, Joost van de Weijer, Laura Lopez Fuentes, et al. Class-balanced active learning for image classification. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 00:3707–3716, 2022. doi: 10.1109/wacv51458.2022.00376.

[15] Matko Bošnjak, Pierre H Richemond, Nenad Tomasev, et al. Semppl: Predicting pseudo-labels for better contrastive representations. *arXiv*, 2023. doi: 10.48550/arxiv.2301.05158.

[16] Jingyao Li, Pengguang Chen, Shaozuo Yu, et al. Bal: Balancing diversity and novelty for active learning, 2023. URL https://arxiv.org/abs/2312.15944.

[17] Marco A.F. Pimentel, David A. Clifton, Lei Clifton, et al. A review of novelty detection. *Signal Processing*, 99:215–249, 2014. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2013.12.026. URL https://www.sciencedirect.com/science/article/pii/S016516841300515X.

[18] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

[19] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, et al. Large-scale long-tailed recognition in an open world. *arXiv*, 2019. doi: 10.48550/arxiv.1904.05160.

[20] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, et al. Open long-tailed recognition in a dynamic world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP (99):1–15, 2022. ISSN 0162-8828. doi: 10.1109/tpami.2022.3200091.

[21] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. *arXiv*, 2021. doi: 10.48550/arxiv.2104.02939.

[22] Kun-Peng Ning, Xun Zhao, Yu Li, et al. Active learning for open-set annotation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 00: 41–49, 2022. doi: 10.1109/cvpr52688.2022.00014.

[23] Jaya Krishna Mandivarapu, Blake Camp, and Rolando Estrada. Deep active learning via open-set recognition. *Frontiers in Artificial Intelligence*, 5:737363, 2022. doi: 10.3389/frai.2022.737363.

[24] Bardia Safaei, Vibashan VS, Celso M. de Melo, et al. Entropic open-set active learning, 2023. URL https://arxiv.org/abs/2312.14126.

[25] Yang Yang, Yuxuan Zhang, XIN SONG, et al. Not all out-of-distribution data are harmful to open-set active learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 13802–13818. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2c8d9636f74d0207ff4f65956010f450-Paper-Conference.pdf.

[26] Zizheng Yan, Delian Ruan, Yushuang Wu, et al. Contrastive open-set active learning based sample selection for image classification. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, PP, 09 2024. doi: 10.1109/TIP.2024.3451928.

[27] Sai Keerthana Goruganthu, Roland R. Oruche, and Prasad Calyam. Adaptive open-set active learning with distance-based out-of-distribution detection for robust task-oriented dialog system. In Tatsuya Kawahara, Vera Demberg, Stefan Ultes, Koji Inoue, Shikib Mehri, David Howcroft, and Kazunori Komatani, editors, *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 357–369, Kyoto, Japan, September 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.sigdial-1.32. URL https://aclanthology.org/2024.sigdial-1.32.

[28] Linhao Qu, Yingfan Ma, Zhiwei Yang, et al. Openal: An efficient deep active learning framework for open-set pathology image classification. *arXiv*, 2023. doi: 10.48550/arxiv.2307.05254.

[29] Abhijit Bendale and Terrance Boult. Towards open set deep networks. *arXiv*, 2015. doi: 10.48550/arxiv.1511.06233.

[30] Dan Hendrycks, Steven Basart, Mantas Mazeika, et al. Scaling out-of-distribution detection for real-world settings. *arXiv*, 2019. doi: 10.48550/arxiv.1911.11132.

[31] Weitang Liu, Xiaoyun Wang, John D Owens, et al. Energy-based out-of-distribution detection. *arXiv*, 2020. doi: 10.48550/arxiv.2010.03759.

[32] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. *arXiv*, 2020. doi: 10.48550/arxiv.2012.06575.

[33] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *arXiv*, 2021. doi: 10.48550/arxiv.2111.12797.

[34] Yiyou Sun, Yifei Ming, Xiaojin Zhu, et al. Out-of-

distribution detection with deep nearest neighbors. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20827–20840. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/sun22d.html.

[35] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, et al. Extremely simple activation shaping for out-of-distribution detection. *arXiv*, 2022. doi: 10.48550/arxiv.2209.09858.

[36] Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. *arXiv*, 2021. doi: 10.48550/arxiv.2111.09805.

[37] Jinsong Zhang, Qiang Fu, Xu Chen, et al. Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KkazG4lgKL.

[38] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv*, 2017. doi: 10.48550/arxiv.1706.02690. ODIN Through input pre-processing (and tempature scaling), determine with classification logits if image is OOD.

[39] Jie Ren, Stanislav Fort, Jeremiah Liu, et al. A simple fix to mahalanobis distance for improving near-ood detection, 2021.

[40] Haoqi Wang, Zhizhong Li, Litong Feng, et al. Vim: Out-of-distribution with virtual-logit matching. *arXiv*, 2022. doi: 10.48550/arxiv.2203.10807.

[41] Songsong Tian, Lusi Li, Weijun Li, et al. A survey on few-shot class-incremental learning. *Neural Networks*, 169:307–324, 2024. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2023.10.039. URL https://www.sciencedirect.com/science/article/pii/S0893608023006019.

[42] Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, et al. Long-tailed class incremental learning, 2022.

[43] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, et al. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–20, 2024. ISSN 1939-3539. doi: 10.1109/tpami.2024.3429383. URL http://dx.doi.org/10.1109/TPAMI.2024.3429383.

[44] Hanbin Zhao, Hui Wang, Yongjian Fu, et al. Memory efficient class-incremental learning for image classification, 2021. URL https://arxiv.org/abs/2008.01411.

[45] Chenze Shao and Yang Feng. Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation, 2022. URL https://arxiv.org/abs/2203.03910.

[46] Flood Sung, Yongxin Yang, Li Zhang, et al. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[47] Bin Wang and Dian Wang. Plant leaves classification: A few-shot learning method based on siamese network. *IEEE Access*, 7:151754–151763, 2019. doi: 10.1109/ACCESS.2019.2947510.

[48] Xiaokang Zhou, Wei Liang, Shohei Shimizu, et al. Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 17(8):5790–5798, 2021. doi: 10.1109/TII.2020.3047675.

[49] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

[50] Aoxue Li, Tiange Luo, Zhiwu Lu, et al. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[51] Zhipeng Lin, Wenjing Yang, Haotian Wang, et al. Scaling few-shot learning for the open world. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(12):13846–13854, Mar. 2024. doi: 10.1609/aaai.v38i12.29291. URL https://ojs.aaai.org/index.php/AAAI/article/view/29291.

[52] Kaiming He, Haoqi Fan, Yuxin Wu, et al. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[53] Xinlei Chen, Haoqi Fan, Ross Girshick, et al. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[54] Hong Liu, Jeff Z. HaoChen, Adrien Gaidon, et al. Self-supervised learning is more robust to dataset imbalance. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=4AZz9osqrar.

[55] Huiwon Jang, Hankook Lee, and Jinwoo Shin. Unsupervised meta-learning via few-shot pseudo-supervised contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TdTGGj7fYYJ.

[56] Anish Acharya, Sujay Sanghavi, Li Jing, et al. Positive unlabeled contrastive learning, 2024. URL https://arxiv.org/abs/2206.01206.

[57] Adam Paszke, Sam Gross, Soumith Chintala, et al. Automatic differentiation in pytorch. 2017.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

[59] Jia Deng, Wei Dong, Richard Socher, et al. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*,

pages 248–255. Ieee, 2009.

[60] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10:1–21, 03 2015. doi: 10.1371/journal.pone.0118432. URL https://doi.org/10.1371/journal.pone.0118432.

[61] Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. Pytorch-ood: A library for out-of-distribution detection based on pytorch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4351–4360, June 2022.

[62] L E Hogeweg, R Gangireddy, D Brunink, V J Kalkman, L Cornelissen, and J W Kamminga. Cood: Combined out-of-distribution detection using multiple measures for anomaly & novel class detection in large-scale hierarchical classification. *arXiv*, 2024.

[63] Ting Chen, Simon Kornblith, Mohammad Norouzi, et al. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

[64] Jean-Bastien Grill, Florian Strub, Florent Altché, et al. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL https://arxiv.org/abs/2006.07733.

[65] Lightly. Lightlyssl is a computer vision framework for self-supervised learning. URL https://github.com/lightly-ai/lightly.

[66] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

[67] Makram Chahine, Ramin Hasani, Patrick Kao, Aaron Ray, Ryan Shubert, Mathias Lechner, Alexander Amini, and Daniela Rus. Robust flight navigation out of distribution with liquid neural networks. *Science Robotics*, 8(77):eadc8892, 2023. doi: 10.1126/scirobotics.adc8892. URL https://www.science.org/doi/abs/10.1126/scirobotics.adc8892.

[68] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

[69] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. URL https://arxiv.org/abs/1602.04938.

APPENDIX

DATASET	TRAIN TIME [HH:MM]	TOP-1 ACC. [%]	TOP-5 ACC. [%]
Places365-LT	13:26	53.5	78.5
ImageNet-LT	26:03	66.4	82.0
iNaturalist2018-Plantae	31:10	63.3	80.7

TABLE A.3: MoCo [52, 53] unsupervised representation learning results. The training time is the total time it took to train the model on the unlabeled dataset. The top-1 and top-5 accuracy are the performance metrics on the respective datasets. Comparatively, the authors of MoCo [52, 53] report a top-1 accuracy of 71.1% on full ImageNet with the same model (ResNet50).

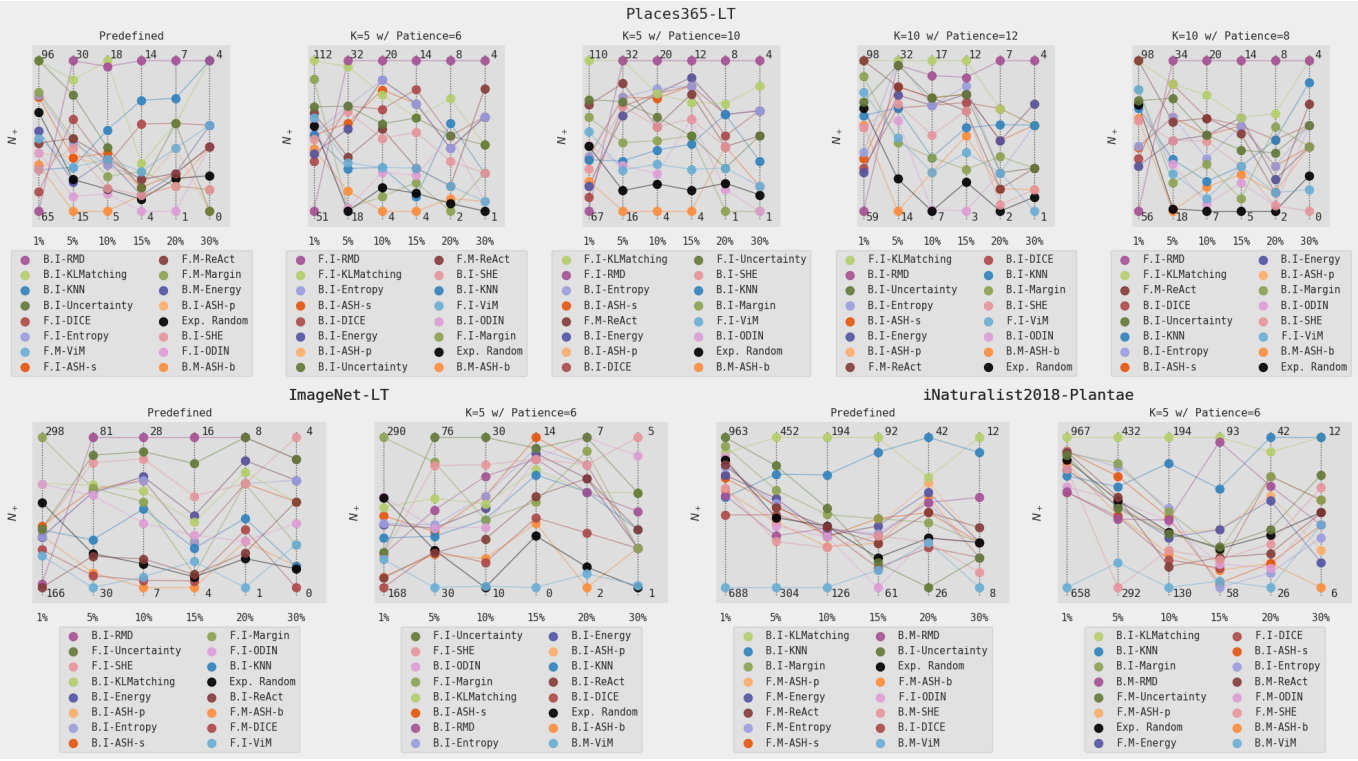

Fig. A.9: Impact of using the predefined validation set versus various configurations of K-Fold cross-validation on novel class detection performance and overfitting on the datasets. The x-axis indicates the percentage of the dataset that is labeled, while the y-axis shows the number of Queried Novel Classes (N_+) ↑.

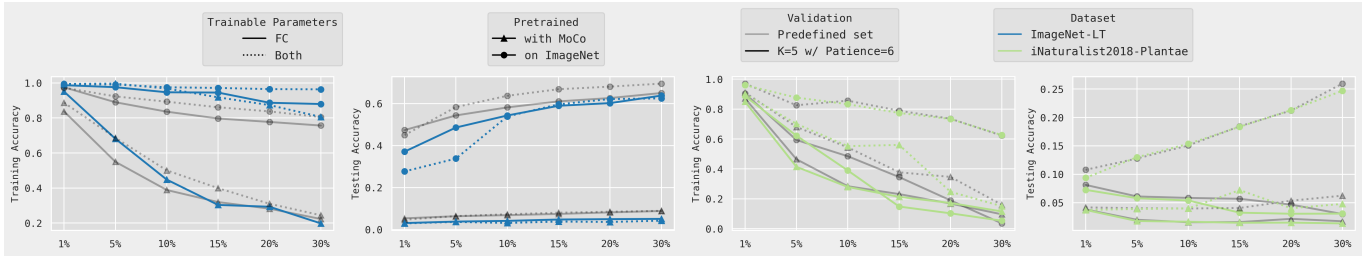

Fig. A.10: Training accuracy ↑ and testing accuracy ↑ using K-Fold cross-validation on the datasets. The x-axis indicates the percentage of the dataset that is labeled, and the y-axis shows the accuracy.

DATASET	$\lvert D_+ \rvert$	D_L	$\lvert N_L \rvert @ D_I$	90%	95%	99%	100%
Places365-LT 62,500 samples 365 classes	1,500	36.0%	70.3%	5,217	8,645	19,346	32,936
ImageNet-LT 115,846 samples 1,000 classes	2,000	19.2%	66.0%	7,361	12,143	27,583	60,797
iNaturalist2018-Plantae 112,966 samples 2,917 classes	3,000	39.8%	46.5%	14,880	20,438	34,423	71,870

TABLE A.4: Configurations of Novelty Detection with Active Class-Incremental Learning on the different datasets. D_+ is the number of samples in each query, and the total number of queries is 15 (including the query of D_I). D_L shows the percentage of the dataset that is labeled at the end of the process. $\lvert N_L \rvert @ D_I$ represents the class coverage at the start of the process. The last four columns show how many samples are required to achieve milestone class coverages when picking samples at random. These values were obtained by running a simulation of the process 500 times with random seeds.

Detector	RMD	KLMatching	Margin	KNN	Uncertainty	Entropy	ASH-s	ODIN	Energy	ASH-p	Random	SHE	DICE	ReAct	ASH-b	ViM
KLMatching	1.7	0.0	0.4	-0.6	-0.8	-1.4	-1.5	-1.0	-1.9	-1.9	-1.1	-2.2	-2.5	-3.4	-3.5	-4.9
Entropy	2.0	1.0	1.2	-0.4	0.9	0.0	-0.6	-0.5	-1.1	-1.1	-0.4	-2.7	-2.6	-2.4	-3.7	-4.7
Margin	1.3	0.1	0.0	-0.7	0.7	0.1	-0.2	-1.2	-0.4	-0.4	-1.0	-0.4	-1.4	-2.1	-3.1	-5.1
Uncertainty	3.3	0.7	1.4	0.9	0.0	-0.2	-1.4	-0.4	-1.8	-1.8	-0.4	-2.2	-1.9	-1.7	-3.7	-2.4
ASH-p	2.8	1.8	1.7	1.7	1.2	0.8	0.3	0.7	0.0	0.0	0.3	-0.1	-0.8	-2.2	-3.1	-4.1
KNN	1.0	1.3	2.0	0.0	1.2	0.4	0.4	0.4	-0.2	-0.2	0.5	0.4	0.6	-0.6	-0.9	-4.5
Energy	4.2	2.3	1.9	2.7	1.1	0.5	0.3	-0.4	0.0	0.0	0.2	-0.8	-1.8	-1.3	-2.6	-3.5
ASH-s	4.4	1.6	2.0	1.0	0.8	0.1	0.0	1.0	0.1	0.1	0.3	-1.0	-0.4	-1.9	-2.4	-2.5
RMD	0.0	2.2	1.8	0.4	2.5	1.9	1.5	1.4	1.8	1.8	-0.3	0.9	0.1	-1.1	-1.6	-4.7
ODIN	5.1	4.2	2.7	3.6	3.0	2.4	1.9	0.0	1.6	1.6	1.1	1.3	0.8	0.8	-1.2	-2.3
SHE	5.8	3.6	2.8	3.7	3.1	2.9	2.2	1.3	1.7	1.7	1.7	0.0	1.2	-0.2	-1.8	-2.5
ReAct	4.1	3.4	3.1	4.2	3.2	3.0	2.9	2.2	2.6	2.6	1.4	2.3	-0.1	0.0	-2.1	-2.3
DICE	4.3	4.6	3.6	4.6	3.9	3.4	3.3	2.8	3.0	3.0	1.9	2.3	0.0	0.8	-0.7	-2.9
ASH-b	6.6	7.3	5.8	6.9	6.9	6.3	5.8	4.5	5.2	5.2	4.0	4.3	4.0	2.6	0.0	0.5
ViM	7.3	7.1	6.7	7.5	7.0	6.5	6.3	6.5	5.8	5.8	4.4	5.6	3.2	4.0	2.5	0.0

(a) **B**oth **C**ontinuous using the **I**mageNet pretrained model

Detector	KLMatching	RMD	KNN	ReAct	Margin	Random	Uncertainty	Entropy	ASH-p	Energy	ASH-s	ASH-b	DICE	SHE	ODIN	ViM
KLMatching	0.0	-0.4	-1.3	-0.3	-0.7	-1.0	-1.4	-1.5	-1.4	-1.8	-1.4	-1.8	-3.7	-2.5	-2.3	-3.4
Uncertainty	1.1	2.1	2.1	0.4	1.3	0.3	0.0	-0.7	-0.9	-1.0	-1.1	-1.6	-1.8	-2.2	-2.0	-1.4
Entropy	2.6	1.7	2.4	0.4	1.7	1.4	-0.2	0.0	-0.8	-0.6	-0.9	-1.3	-2.1	-1.9	-2.4	-0.4
ASH-s	1.9	2.6	2.0	0.8	0.9	1.3	0.2	0.4	-0.3	-0.2	0.0	-1.2	-2.2	-1.9	-2.1	-1.9
ReAct	1.4	2.1	2.0	0.0	1.1	1.4	-0.4	0.1	0.1	0.1	0.0	-0.4	-0.7	-2.0	-1.7	-2.1
Margin	1.9	0.8	1.3	1.6	0.0	-0.7	0.2	0.7	0.3	0.3	0.6	-0.1	0.2	-1.3	-1.9	-1.3
Energy	2.2	3.4	2.8	1.5	1.4	1.6	0.6	0.3	-0.1	0.0	-0.0	-0.6	-0.8	-1.6	-1.6	-0.8
ASH-p	3.4	3.5	2.5	1.3	1.6	1.9	0.8	0.4	0.0	0.0	0.3	-0.1	-1.3	-1.2	-1.5	-0.5
KNN	3.8	1.9	0.0	2.8	2.2	1.6	2.7	2.6	2.1	1.9	1.9	1.4	0.3	0.4	0.1	-2.1
ASH-b	3.5	3.7	3.1	2.5	3.1	2.8	2.1	1.9	1.2	1.3	1.1	0.0	1.6	-0.4	-0.1	0.8
RMD	3.8	0.0	0.2	3.6	1.7	1.4	2.9	3.4	3.6	3.3	3.0	2.6	1.4	0.4	1.4	-0.4
SHE	5.7	5.4	4.2	4.0	3.1	3.4	3.4	3.6	3.2	3.3	2.6	1.7	2.0	0.0	1.0	2.7
ODIN	5.5	4.2	4.7	4.6	4.2	3.5	4.6	4.2	4.0	4.1	3.9	3.2	1.7	2.1	0.0	1.6
DICE	6.4	5.9	5.3	4.4	4.4	4.3	4.1	4.3	3.9	3.9	3.7	3.1	0.0	1.3	1.4	0.3
ViM	6.9	6.3	6.4	5.8	5.0	4.6	5.4	5.1	4.8	4.8	4.8	4.5	2.2	3.6	3.3	0.0

(b) **B**oth **C**ontinuous using the **M**oCo pretrained model

Detector	RMD	KLMatching	Margin	Uncertainty	Random	KNN	ASH-s	SHE	Entropy	ODIN	DICE	ASH-p	Energy	ViM	ASH-b	ReAct
Margin	4.6	1.8	0.0	0.4	-0.4	-0.2	-0.2	-0.0	-0.6	-1.3	-0.3	-1.1	-1.1	-1.8	-2.4	-1.8
Uncertainty	3.8	0.8	1.8	0.0	0.2	0.1	-0.9	0.1	-0.9	-0.3	-0.2	-0.8	-0.8	-1.3	-1.1	-2.2
RMD	0.0	0.9	2.7	1.2	1.0	0.2	0.7	0.2	0.6	1.1	-1.3	0.3	0.3	-2.4	0.1	-0.8
KLMatching	2.0	0.0	2.2	1.8	0.4	-1.4	1.3	0.4	1.0	0.4	0.4	1.1	1.1	-2.4	1.6	-0.1
Entropy	4.7	3.8	3.1	2.8	1.8	3.1	-0.5	0.1	0.0	0.6	-1.3	-1.6	-1.6	0.8	-1.5	-1.8
SHE	5.4	2.9	2.9	2.4	1.5	0.7	0.6	0.0	0.6	0.6	0.9	0.2	0.2	-0.3	-0.0	-0.7
ASH-s	5.2	4.3	3.4	2.3	2.4	3.0	0.0	0.5	0.8	0.4	-0.1	-1.6	-1.6	1.1	-0.4	-0.8
ODIN	6.1	3.7	3.0	2.0	2.2	1.5	0.6	0.6	0.2	0.0	1.3	-0.0	-0.0	1.0	0.1	-0.7
KNN	5.6	3.2	3.3	2.6	2.3	0.0	2.3	2.4	2.1	2.5	2.8	2.1	2.1	-2.0	2.1	-0.5
Energy	5.9	5.0	4.3	3.2	3.1	4.2	2.1	2.0	2.3	2.2	2.2	0.3	0.0	1.9	-0.5	-0.5
ASH-p	5.6	5.3	3.8	3.4	3.4	4.2	2.4	1.7	2.6	1.9	0.5	0.0	0.0	1.5	0.0	-0.1
ReAct	5.4	5.2	3.9	4.2	2.9	4.3	2.8	2.8	2.8	2.6	0.9	1.4	1.4	2.6	-1.5	0.0
DICE	5.6	6.4	4.9	4.9	3.7	5.0	3.5	4.2	3.4	4.0	0.0	2.5	2.5	2.9	-0.0	1.6
ViM	7.3	4.7	4.9	4.5	3.3	1.5	4.5	4.3	4.0	3.6	3.7	3.9	3.9	0.0	2.8	2.4
ASH-b	7.7	6.9	5.3	6.3	4.0	5.5	5.3	4.9	5.1	4.2	3.3	4.0	4.0	4.0	0.0	2.6

(c) **B**oth **R**eload using the **I**mageNet pretrained model

Detector	KLMatching	Random	RMD	Margin	ReAct	KNN	ViM	Uncertainty	Energy	ASH-p	Entropy	SHE	ASH-s	ASH-b	DICE	ODIN
Margin	2.2	0.1	-0.6	0.0	1.0	-1.6	-2.2	-1.4	0.3	0.4	-0.5	-1.5	-0.9	-1.1	-2.5	-1.5
KLMatching	0.0	0.8	-0.8	1.4	2.9	-3.3	-2.4	0.5	0.6	0.5	0.7	-0.3	0.4	1.0	1.0	-0.9
ReAct	4.2	4.0	2.0	3.7	0.0	1.8	0.9	0.6	-0.1	-0.1	0.1	2.2	-0.4	0.1	-0.3	-0.8
Uncertainty	6.0	4.3	4.7	3.5	2.6	3.7	3.0	0.0	-0.6	-0.6	-0.4	0.7	-1.1	-1.5	-1.2	-0.9
RMD	2.2	3.0	0.0	2.8	3.2	-2.1	-3.0	3.9	3.2	3.4	3.7	1.5	3.5	2.1	0.6	1.7
Energy	6.1	4.4	4.6	3.8	3.5	5.7	4.7	-0.0	0.0	-0.1	-0.2	0.9	-0.6	-0.2	-1.2	-0.6
Entropy	6.4	4.7	4.7	4.2	2.9	5.0	4.2	0.7	0.1	0.1	0.0	1.2	-0.6	-0.4	-0.8	-0.6
ASH-p	6.6	4.6	4.3	4.4	3.7	4.8	4.2	0.6	0.2	0.0	0.1	1.0	-0.4	-0.2	-0.8	-0.6
ASH-s	6.3	4.2	4.9	3.5	3.6	5.0	4.7	1.1	0.7	0.5	0.3	0.8	0.0	0.5	-0.6	-0.3
ASH-b	6.7	4.9	5.0	4.3	4.2	4.9	4.3	0.7	0.3	0.2	0.1	0.9	-0.3	0.0	-0.2	-0.2
ODIN	6.8	4.9	5.4	4.2	4.0	4.9	4.5	0.5	0.7	0.6	-0.1	0.9	-0.1	0.1	0.1	0.0
SHE	5.5	4.2	5.1	4.1	4.9	2.8	2.8	2.1	2.3	2.2	1.9	0.0	1.9	2.4	2.2	0.5
DICE	7.2	5.4	5.6	4.5	4.8	5.1	5.0	1.2	1.3	1.4	1.2	1.8	0.7	0.6	0.0	0.5
KNN	4.2	3.9	3.3	4.1	5.6	0.0	-1.5	5.3	5.9	5.8	5.9	3.2	6.1	5.2	4.1	4.0
ViM	5.8	6.0	6.1	5.7	6.5	3.1	0.0	6.5	6.8	6.7	6.6	4.7	7.0	6.2	6.1	4.5

(d) **B**oth **R**eload using the **M**oCo pretrained model

Detector	RMD	KLMatching	Margin	Uncertainty	Random	ASH-s	KNN	SHE	Entropy	ODIN	DICE	ASH-p	Energy	ViM	ReAct	ASH-b
Margin	2.8	0.1	0.0	-0.3	-1.0	-1.6	-1.5	-1.3	-2.2	-1.7	-1.9	-2.5	-2.5	-2.8	-3.1	-3.6
RMD	0.0	-1.1	2.0	0.1	0.5	0.0	-2.4	-0.1	0.0	-0.6	-1.9	0.4	0.4	-4.4	-0.9	-0.6
Uncertainty	4.0	1.4	1.1	0.0	1.1	-1.4	0.9	-0.4	-1.5	-0.6	-0.0	-1.7	-1.7	-0.8	-2.3	-1.3
KLMatching	3.6	0.0	1.1	1.0	0.3	0.7	-1.5	0.4	0.3	0.5	0.5	0.1	0.1	-2.1	-0.9	1.0
ASH-s	4.7	3.0	2.3	1.0	1.6	0.0	1.2	0.1	0.1	-0.5	-0.8	-1.7	-1.7	0.4	-1.5	-1.4
Entropy	4.5	2.7	2.3	1.2	1.2	-0.0	1.6	0.4	0.0	-0.1	0.4	-1.4	-1.4	0.4	-1.0	-1.2
SHE	5.4	3.4	2.4	2.9	1.6	1.1	0.6	0.0	0.9	0.1	1.2	0.3	0.3	-0.8	-0.5	-0.1
ODIN	6.9	4.2	3.0	2.2	2.1	0.6	2.3	1.7	0.3	0.0	1.4	-0.2	-0.2	1.2	-0.5	-0.2
ASH-p	4.9	4.3	3.3	2.3	2.7	2.1	2.1	0.9	2.1	1.3	-0.2	0.0	0.0	0.7	-0.4	-0.3
Energy	5.3	3.5	3.6	2.6	2.3	2.1	1.7	1.1	2.0	1.2	0.3	0.0	0.0	0.9	-0.0	-0.2
ReAct	5.7	4.2	3.7	3.0	2.9	2.0	3.2	1.5	1.9	1.6	1.3	0.8	0.8	1.8	0.0	-0.7
DICE	6.3	6.0	4.5	4.1	2.9	2.8	4.3	3.2	2.4	2.5	0.0	1.9	1.9	2.0	1.0	-0.6
KNN	6.2	3.8	4.0	3.4	2.9	3.4	0.0	2.5	3.1	3.2	3.0	3.1	3.1	-1.2	1.8	3.4
ViM	7.0	5.4	5.4	4.9	3.4	4.5	1.4	3.9	4.2	3.6	3.7	4.1	4.1	0.0	2.8	2.7
ASH-b	7.5	6.7	5.0	5.7	3.6	4.7	5.3	4.5	4.5	4.0	3.0	3.6	3.6	3.9	2.5	0.0

(e) **F**rozen using the **I**mageNet pretrained model

Detector	KLMatching	Random	RMD	Margin	ReAct	KNN	ViM	ASH-p	Energy	Uncertainty	Entropy	ASH-s	ODIN	SHE	ASH-b	DICE
Margin	2.0	0.1	-0.7	0.0	-0.7	-0.7	-1.3	-1.9	-2.4	-1.5	-2.1	-2.8	-1.8	-2.5	-3.5	-2.9
KLMatching	0.0	0.0	-1.3	0.4	-0.1	-2.9	-4.6	-1.2	-1.4	-0.4	-1.3	-1.7	-0.4	-2.5	-2.7	-1.7
ReAct	4.9	3.9	3.9	3.5	0.0	1.8	0.7	-0.4	-0.4	0.8	-0.1	-0.5	0.3	0.8	-1.1	-1.1
RMD	1.6	1.4	0.0	1.7	3.9	-2.4	-3.5	3.3	3.2	2.7	3.9	3.2	0.4	-0.2	1.8	0.0
Uncertainty	5.9	3.3	4.0	2.8	3.2	3.7	3.0	0.6	0.6	0.0	-0.4	-0.6	-0.5	0.6	-0.8	-1.1
ASH-p	5.7	4.1	3.9	3.4	3.0	4.4	3.4	0.0	0.1	0.1	-0.4	-0.6	-0.1	0.4	-0.3	-0.8
Energy	6.1	4.2	3.7	4.2	3.7	4.0	3.3	-0.1	0.0	0.3	-0.5	-0.8	0.1	1.0	-0.7	-1.2
Entropy	6.4	4.6	4.8	3.7	3.4	4.1	3.8	0.3	0.4	0.4	0.0	-0.5	0.3	1.0	-0.4	-0.6
ASH-s	6.4	4.6	5.0	4.1	3.5	4.9	3.2	0.8	0.9	0.6	0.0	0.0	0.0	0.5	-0.2	-0.6
ASH-b	6.3	5.4	4.0	4.1	4.5	4.3	4.4	0.6	0.7	0.7	0.2	-0.2	0.7	1.0	0.0	-0.4
ODIN	6.0	4.1	4.6	4.1	4.5	3.8	2.8	1.5	1.5	1.3	1.3	0.5	0.0	0.1	0.3	0.1
SHE	5.6	4.2	4.9	4.1	4.5	2.3	2.2	2.0	2.2	1.4	1.5	1.5	0.4	0.0	1.8	1.5
DICE	7.0	5.5	6.2	4.5	4.2	5.1	5.8	1.3	1.4	1.0	0.9	0.8	1.1	1.5	0.9	0.0
KNN	4.1	3.4	3.1	4.0	5.1	0.0	-2.1	4.9	5.0	4.5	5.0	5.1	3.9	2.0	4.0	3.3
ViM	5.4	5.1	5.6	5.6	6.3	2.5	0.0	7.0	7.1	5.4	6.6	7.1	4.7	4.0	6.2	5.7

(f) **F**rozen using the **M**oCo pretrained model

TABLE A.5: These tables illustrate the total number of potentially novel classes discovered by alternative detectors (columns) compared to the active detector (rows) throughout the entire annotation process across all datasets. The values are normalized per column to mitigate bias towards earlier queries, where more novel classes are typically present. Each value represents the sum of normalized potential novel class detections across the entire annotation process. A positive value indicates that the alternative detector would have discovered additional novel classes compared to the active detector. If a row predominantly contains negative values, it suggests that the active detector outperforms most alternative detectors. The active detectors are sorted by their normalized average number of potentially missed novel classes, with the top row representing the best-performing detector for these configurations. Conversely, the columns are sorted based on the average normalized number of potentially discovered novel classes, with the left column representing the best-performing alternative detector. Note that this analysis does not account for the iterative nature of the process, where a detector might identify the same novel class multiple times.

Pretrained	Detector	Dataset Class Coverage Config	Places365-LT 90%	95%	99%	100%	ImageNet-LT 90%	95%	99%	100%	iNaturalist2018-Plantae 90%	95%	99%	100%
ImageNet	Committee	Both Reload	-36.4%	**-48.8%**	**-60.2%**	-	**-31.3%**	**-45.1%**	**-59.2%**	-	-12.5%	-11.6%	-7.6%	-
		Frozen	-35.6%	-46.9%	**-63.6%**	-	-29.8%	**-44.0%**	**-62.0%**	**-57.6%**	-10.5%	-8.4%	-11.2%	-
	Committee*	Both Reload	-37.7%	-38.4%	-47.4%	-	**-30.7%**	-43.8%	-58.0%	**-57.6%**	-12.4%	-10.6%	-10.9%	-
		Frozen	-36.1%	-41.2%	-48.4%	**-59.1%**	-30.3%	**-44.2%**	-56.2%	-	-5.6%	-3.6%	-4.6%	-
	KLMatching	Both Reload	-38.5%	**-48.8%**	-56.5%	**-46.9%**	-24.7%	-31.2%	-36.5%	-	-14.3%	-7.7%	-1.4%	-
	Uncertainty	Both Cont.	**-39.8%**	-43.8%	-51.4%	-	-21.5%	-28.7%	-28.7%	-	-18.4%	-19.2%	-6.1%	-
	Committee	Both Cont.	**-39.4%**	**-50.6%**	-46.4%	-	-17.3%	-25.4%	-25.1%	-	**-20.0%**	**-20.7%**	**-20.9%**	-
	Committee*	Both Cont.	-33.8%	-38.4%	-50.9%	-	-22.1%	-22.8%	-32.2%	-	-16.2%	-15.2%	-18.2%	-
	KLMatching	Frozen	-36.1%	-44.9%	**-58.5%**	-	-18.4%	-26.0%	-33.1%	-	-12.5%	-5.4%	-1.8%	-
	Entropy	Both Cont.	-36.0%	-41.8%	-51.1%	**-46.9%**	-17.4%	-18.0%	-24.2%	-	-14.5%	-11.1%	-5.1%	-
	Margin	Both Cont.	-32.5%	-36.5%	-45.4%	-	-24.9%	-21.2%	-22.7%	-	**-21.0%**	**-20.5%**	**-22.8%**	-
	Uncertainty	Both Reload	**-41.2%**	-44.6%	-35.0%	-	**-30.4%**	-43.1%	**-59.9%**	-	-2.6%	+6.7%	+12.0%	-
	KLMatching	Both Cont.	-32.1%	-34.4%	-36.8%	-	-20.5%	-20.3%	-24.7%	-	**-25.5%**	**-25.1%**	**-23.0%**	-
	RMD	Frozen	-31.2%	-36.6%	-51.9%	**-46.9%**	-17.0%	-36.4%	-55.7%	-	-9.2%	-2.2%	-3.5%	-
	Margin	Both Reload	-28.2%	-23.8%	-10.0%	-	-30.2%	-41.5%	-51.3%	-	-10.6%	-10.2%	-11.6%	-
	RMD	Both Reload	-31.8%	-42.9%	-45.4%	-42.8%	-15.0%	-34.4%	-55.4%	-	-7.7%	-3.0%	-3.8%	-
	Margin	Frozen	-27.9%	-17.2%	-27.9%	-	-30.3%	-42.8%	-55.4%	-	-10.3%	-9.2%	-10.9%	-
	RMD	Both Cont.	-30.3%	-39.1%	-48.7%	-	-1.1%	-15.8%	-30.9%	-	-10.7%	-8.3%	-7.2%	-
	ASH-s	Both Cont.	-34.9%	-38.9%	-48.0%	-	-17.2%	-17.7%	-18.1%	-	-11.4%	-4.7%	-3.6%	-
	KNN	Both Cont.	-28.1%	-38.3%	-43.7%	-	-13.1%	-20.6%	-22.4%	-	-12.1%	-9.5%	-11.3%	-
	Energy	Both Cont.	-34.0%	-38.7%	-47.9%	-42.8%	-14.2%	-22.8%	-9.7%	-	-9.1%	-5.1%	-2.0%	-
	Uncertainty	Frozen	-37.5%	-40.8%	-31.3%	-	-25.3%	-41.0%	-58.2%	-	+4.5%	+12.1%	+9.5%	-
	ASH-p	Both Cont.	-34.5%	-41.3%	-43.7%	-	-12.5%	-15.5%	-21.4%	-	-8.3%	-6.5%	-1.6%	-
	ASH-s	Frozen	-37.4%	-30.2%	-34.6%	-	-24.7%	-33.3%	-43.6%	-	+6.6%	+15.3%	+7.9%	-
	Entropy	Both Reload	-24.1%	-21.8%	-33.7%	-	-20.0%	-29.6%	-32.0%	-	-0.7%	+2.5%	-0.7%	-
		Frozen	-30.7%	-35.3%	-31.9%	-	-26.2%	-30.9%	-42.3%	-	+9.5%	+12.8%	+6.7%	-
	SHE	Both Reload	+1.2%	+5.2%	+0.4%	-	-26.3%	-38.3%	-33.1%	-	-7.4%	-5.1%	-0.7%	-
	ASH-s	Both Reload	-25.2%	-21.9%	-26.7%	-	-25.1%	-31.6%	-38.1%	-	+1.4%	+3.1%	+14.4%	-
	SHE	Both Cont.	-34.6%	-33.1%	-10.0%	-	-13.1%	-14.7%	-8.0%	-	-9.6%	+2.5%	+11.1%	-
		Frozen	+10.7%	+15.2%	-4.1%	-	-21.7%	-36.1%	-25.3%	-	-0.7%	+3.4%	+4.0%	-
	KNN	Both Reload	-29.5%	-23.2%	-36.6%	**-46.9%**	+26.8%	+61.5%	-	-	+0.5%	+3.5%	+3.8%	-
	ODIN	Both Reload	-25.9%	+3.8%	-	-	-29.5%	-40.7%	-25.5%	-	+5.9%	+8.7%	+16.3%	-
		Frozen	-18.1%	+6.8%	-	-	-23.7%	-36.6%	-22.4%	-	-0.8%	+4.3%	+12.1%	-
	ASH-p	Frozen	-26.0%	-22.7%	-12.5%	-	-15.4%	-6.5%	-	-	+4.6%	+9.3%	+6.0%	-
	ODIN	Both Cont.	-24.1%	-14.0%	-29.5%	-	-17.3%	-20.6%	-24.2%	-	+3.2%	+11.7%	-	-
	Energy	Frozen	-23.2%	-18.8%	-17.1%	-	-16.4%	-9.8%	-	-	+4.7%	+5.9%	+6.9%	-
	ReAct	Both Cont.	-25.4%	-23.9%	-43.6%	-42.8%	+9.5%	+12.7%	-	-	+3.1%	+11.0%	-	-
	DICE	Both Cont.	-28.2%	-36.4%	-46.4%	-	-1.6%	+2.5%	-1.3%	-	+23.4%	+35.1%	-	-
	ASH-p	Both Reload	-25.8%	-6.6%	+5.2%	-	-4.9%	+4.4%	-	-	+2.7%	+7.2%	+6.9%	-
	ReAct	Frozen	-22.9%	-14.6%	-18.0%	-	+9.5%	+28.1%	-	-	+5.5%	+6.6%	+7.8%	-
	DICE	Frozen	-24.6%	-26.8%	-30.2%	-	+43.8%	+59.1%	-	-	+3.6%	+10.5%	+10.3%	-
	KNN	Frozen	-19.9%	-17.3%	-29.7%	-38.7%	+26.2%	+57.4%	-	-	+4.8%	+6.6%	+8.6%	-
	Energy	Both Reload	-22.7%	-8.2%	-1.5%	-	-11.3%	-1.3%	-	-	+0.8%	+8.4%	+12.0%	-
	ReAct	Both Reload	-20.1%	-5.1%	-18.0%	-	+10.6%	+24.3%	-	-	+4.6%	+7.0%	+10.2%	-
	DICE	Both Reload	-14.9%	-10.2%	-5.4%	-	+33.2%	+46.1%	-	-	-1.0%	+5.7%	+12.6%	-
	ViM	Both Reload	-2.8%	-8.2%	-27.5%	-	+74.8%	+86.4%	-	-	+7.4%	+13.2%	+10.2%	-
		Frozen	-6.2%	-0.2%	-24.4%	-	+31.1%	+56.1%	-	-	+15.3%	+18.0%	+15.3%	-
		Both Cont.	-1.9%	-4.9%	-24.2%	-	+124.0%	+94.9%	-	-	+46.6%	+75.2%	-	-
MoCo	Committee*	Both Reload	-28.2%	-30.3%	-38.4%	**-42.8%**	**-11.5%**	**-13.3%**	**-21.6%**	-	**-13.7%**	-9.7%	-11.4%	-
		Frozen	-27.6%	-33.6%	**-44.6%**	-38.7%	-5.2%	-12.4%	**-23.1%**	-	-12.6%	-11.0%	**-13.4%**	-
	Committee	Frozen	-28.8%	-32.7%	**-51.4%**	-	**-11.7%**	-12.4%	**-14.7%**	-	-10.3%	-9.3%	-8.3%	-
	KLMatching	Both Cont.	-27.6%	-36.9%	-20.4%	-38.7%	-7.7%	-11.5%	-13.7%	-	**-16.9%**	**-15.6%**	**-15.2%**	-
		Both Reload	-30.1%	-30.2%	-33.2%	-	-10.3%	-9.0%	-	-	-12.3%	**-11.7%**	-4.3%	-
		Frozen	-32.2%	**-38.6%**	-39.3%	**-42.8%**	-4.7%	-1.9%	-4.1%	-	-8.9%	-9.9%	-9.1%	-
	Margin	Frozen	-26.3%	-22.1%	-12.0%	-	**-10.6%**	-11.0%	-4.1%	-	**-13.3%**	**-12.1%**	**-16.9%**	-
	Committee	Both Reload	**-34.8%**	-33.7%	-41.7%	-	-1.6%	-6.0%	-5.4%	-	-11.3%	-8.5%	-3.6%	-
	Margin	Both Cont.	-20.3%	-14.6%	-29.5%	-	-10.1%	**-13.4%**	-8.6%	-	-11.6%	-9.1%	-10.5%	-
	Committee	Both Cont.	-33.3%	-26.7%	-29.7%	-	-0.7%	+0.1%	-	-	-13.2%	-10.5%	-8.0%	-
	Committee*	Both Cont.	-27.5%	-25.0%	-26.1%	-	-5.8%	-9.3%	-	-	-12.0%	-3.3%	-5.1%	-
	Margin	Both Reload	-25.5%	-12.2%	-2.9%	-	-7.1%	**-12.5%**	-3.0%	-	-11.5%	-11.6%	-10.3%	-
	Uncertainty	Both Cont.	-31.7%	-36.5%	**-45.2%**	-38.7%	-8.1%	-7.9%	-	-	-4.1%	+6.9%	-	-
	ReAct	Both Cont.	**-34.7%**	-37.3%	-40.8%	**-42.8%**	-3.1%	-2.4%	-1.0%	-	-4.5%	+7.2%	+12.3%	-
	ASH-p	Both Cont.	-21.8%	-20.6%	-31.3%	-	-7.6%	-3.1%	-	-	-3.9%	+1.5%	+5.8%	-
	KNN	Both Cont.	-23.2%	-20.9%	-23.1%	-	-5.6%	-7.4%	+0.3%	-	-6.2%	-1.0%	+5.8%	-
	ASH-s	Both Cont.	-14.4%	-24.4%	-38.4%	**-42.8%**	-6.2%	+3.1%	-6.4%	-	-1.0%	+6.3%	+8.8%	-
	Entropy	Both Cont.	-29.4%	-34.5%	-32.2%	-	-7.9%	-0.0%	+0.3%	-	+4.8%	+11.0%	-	-
	Energy	Both Cont.	-11.4%	-14.2%	-24.2%	-	-5.0%	-6.3%	-6.4%	-	-1.7%	+3.4%	+10.2%	-
	ReAct	Both Reload	**-34.6%**	**-39.1%**	-25.1%	-	-3.1%	+19.3%	-	-	-2.4%	+9.0%	-	-
	RMD	Both Cont.	-12.7%	-17.8%	-23.1%	-	+3.5%	+2.9%	-4.7%	-	-5.3%	-1.9%	-1.6%	-
	ReAct	Frozen	-28.4%	**-37.5%**	-26.7%	-	-3.6%	+12.3%	-	-	+2.9%	+14.0%	-	-
	RMD	Frozen	-23.0%	-24.2%	-30.2%	-	+5.5%	-2.6%	-10.8%	-	+6.8%	+14.8%	+14.7%	-
		Both Reload	-25.1%	-25.7%	-40.8%	-	+8.8%	+0.8%	-1.6%	-	+7.4%	+15.3%	+12.6%	-
	ASH-b	Both Cont.	-13.4%	-11.4%	-24.2%	-	-4.1%	-1.2%	+0.3%	-	+2.2%	+18.4%	-	-
	Uncertainty	Both Reload	-24.8%	+44.8%	-	-	-3.7%	+19.3%	-	-	-1.7%	+12.0%	-	-
	ASH-p	Frozen	-5.4%	+17.7%	-	-	+14.5%	+43.2%	-	-	-2.6%	+3.5%	-	-
	Uncertainty	Frozen	-4.1%	+67.6%	-	-	+0.1%	+38.2%	-	-	-0.7%	+6.9%	+18.1%	-
	KNN	Frozen	-24.1%	-13.4%	-18.0%	-	+0.1%	+11.0%	-	-	+51.7%	+72.2%	-	-
	DICE	Both Cont.	-24.1%	-5.0%	-1.5%	-	+3.7%	+15.2%	-	-	+12.1%	+20.4%	-	-
	Energy	Both Reload	+22.3%	+28.7%	-	-	+5.5%	+39.2%	-	-	-1.5%	+8.8%	+16.4%	-
	KNN	Both Reload	-27.0%	-14.2%	-16.0%	-	+20.2%	+31.7%	-	-	+49.4%	+62.9%	-	-
	Energy	Frozen	-5.1%	+26.1%	-	-	+16.4%	+41.1%	-	-	-1.2%	+13.5%	-	-
	ASH-p	Both Reload	+13.9%	+45.7%	-	-	+17.6%	+53.7%	-	-	-3.5%	+6.9%	-	-
	Entropy	Both Reload	-3.8%	+78.3%	-	-	+21.5%	+59.9%	-	-	-3.3%	+5.6%	+13.7%	-
	ODIN	Frozen	-3.0%	+85.6%	-	-	+19.2%	+49.1%	-	-	+1.2%	+6.0%	+12.0%	-
	SHE	Both Cont.	+50.3%	+109.1%	-	-	-4.7%	-5.8%	-	-	+8.4%	+16.5%	-	-
		Both Reload	+170.6%	-	-	-	+31.7%	+42.8%	-	-	-0.6%	+5.0%	+3.2%	-
	ODIN	Both Cont.	-7.6%	-2.0%	-	-	+14.2%	+14.3%	-	-	+19.2%	+33.4%	-	-
	ASH-s	Frozen	-5.1%	+41.7%	-	-	+34.6%	+56.9%	-	-	+1.5%	+6.2%	-	-
	ViM	Both Cont.	-10.9%	-6.9%	+0.4%	-	+30.3%	+44.1%	-	-	+5.0%	+18.5%	-	-
		Frozen	-25.3%	-1.2%	-18.2%	-	+40.5%	+46.5%	-	-	+93.6%	-	-	-
		Both Reload	-17.8%	-6.9%	-15.7%	-	+49.7%	+50.6%	-	-	+85.9%	-	-	-

TABLE A.6: Results of Novelty Detection with Active Class-Incremental Learning from Sections IV-C and IV-D. Per dataset and pretrained model, the values represent the percentage of $D_L \downarrow$ required to achieve a certain class coverage compared to random sampling. The detectors are sorted by their overall performance across all datasets based on ranking. The best-performing detector per milestone is highlighted in bold. Detectors that performed worse than random sampling on all milestones across all datasets are excluded from the table, including: On ImageNet, B+BC+F-ASH-b; On MoCo, B+F-ASH-b, B-ASH-s, B+F-DICE, F-Entropy, B-ODIN, and F-SHE.