SIMILARITY SEARCH FOR EFFICIENT ACTIVE LEARNING AND SEARCH OF RARE CONCEPTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many active learning and search approaches are intractable for industrial settings with billions of unlabeled examples. Existing approaches, such as uncertainty sampling or information density, search globally for the optimal examples to label, scaling linearly or even quadratically with the unlabeled data. However, in practice, data is often heavily skewed; only a small fraction of collected data will be relevant for a given learning task. For example, when identifying rare classes, detecting malicious content, or debugging model performance, positive examples can appear in less than 1% of the data. In this work, we exploit this skew in large training datasets to reduce the number of unlabeled examples considered in each selection round by only looking at the nearest neighbors to the labeled examples. Empirically, we observe that learned representations can effectively cluster unseen concepts, making active learning very effective and substantially reducing the number of viable unlabeled examples. We evaluate several selection strategies in this setting on three large-scale computer vision datasets: ImageNet, OpenImages, and a proprietary dataset of 10 billion images from a large internet company. For rare classes, active learning methods need as little as 0.31% of the labeled data to match the average precision of full supervision. By limiting the selection strategies to the immediate neighbors of the labeled data as candidates for labeling, we process as little as 0.1% of the unlabeled data while achieving similar reductions in labeling costs as the traditional global approach. This process of expanding the candidate pool with the nearest neighbors of the labeled set can be done efficiently and reduces the computational complexity of selection by orders of magnitude.

1 INTRODUCTION

Large-scale unlabeled datasets contain millions or billions of examples spread over a wide variety of underlying concepts (Chelba et al., 2013; Zhu et al., 2015; Zhang et al., 2015; Wan et al., 2019; Russakovsky et al., 2015; Kuznetsova et al., 2020; Thomee et al., 2016; Abu-El-Haija et al., 2016; Caesar et al., 2019; Lee et al., 2019). Often, these massive datasets skew towards a relatively small number of common concepts, such as cats, dogs, and people (Liu et al., 2019; Zhang et al., 2017; Wang et al., 2017; Van Horn & Perona, 2017). Rare concepts, such as harbor seals, may only appear in a small fraction of the data (less than 1%). However, in many settings, performance on these rare concepts is critical. For example, harmful or malicious content may comprise a small percentage of user-generated content, but it can have an outsize impact on the overall user experience (Wan et al., 2019). Similarly, when debugging model behavior for safety-critical applications like autonomous vehicles, or when dealing with representational biases in models, obtaining data that captures rare concepts allows machine learning practitioners to combat blind spots in model performance (Karpathy, 2018; Holstein et al., 2019; Ashmawy et al., 2019; Karpathy, 2020). Even a simple prediction task like stop sign detection can be challenging given the diversity of real-world data. Stop signs may appear in a variety of conditions (e.g., on a wall or held by a person), be heavily occluded, or have modifiers (e.g., "Except Right Turns") (Karpathy, 2020). While large-scale datasets are core to addressing these issues, finding the relevant examples for these long-tail tasks is challenging.

Active learning and search have the potential to automate the process of identifying these rare, high value data points significantly, but existing methods become intractable at this scale. Specifically, the goal of active learning is to reduce the cost of labeling (Settles, 2012). To this end, the learning algorithm is allowed to choose which data to label based on uncertainty (e.g., the entropy of

predicted class probabilities) or other heuristics (Settles, 2011; 2012; Lewis & Gale, 1994). Active search is a sub-area focused on finding positive examples in skewed distributions (Garnett et al., 2012). Because of a concentrated focus on labeling costs, existing techniques, such as uncertainty sampling (Lewis & Gale, 1994) or information density (Settles & Craven, 2008), perform multiple selection rounds and iterate over the entire unlabeled data to identify the optimal example or batch of examples to label and scale linearly or even quadratically with the size of the unlabeled data. Computational efficiency is becoming an impediment as the size of datasets and model complexities have increased (Amodei & Hernandez, 2018). Recent work has tried to address this problem with sophisticated methods to select larger and more diverse batches of examples in each selection round and reduce the total number of rounds needed to reach the target labeling budget (Sener & Savarese, 2018; Kirsch et al., 2019; Coleman et al., 2020; Pinsler et al., 2019; Jiang et al., 2018). Nevertheless, these approaches still scan over all of the examples to find the optimal examples to label in each round and can be intractable for large-scale unlabeled datasets. For example, running a single inference pass over 10 billion images with ResNet-50 (He et al., 2016) would take 38 exaFLOPs.

In this work, we propose Similarity search for Efficient Active Learning and Search (SEALS) to restrict the candidates considered in each selection round and vastly reduce the computational complexity of active learning and search methods. Empirically, we find that learned representations from pre-trained models can effectively cluster many unseen and rare concepts. We exploit this latent structure to improve the computational efficiency of active learning and search methods by only considering the nearest neighbors of the currently labeled examples in each selection round. This can be done transparently for many selection strategies making SEALS widely applicable. Finding the nearest neighbors for each labeled example in unlabeled data can be performed efficiently with sublinear retrieval times (Charikar, 2002) and sub-second latency on billion-scale datasets (Johnson et al., 2017) for approximate approaches. While constructing the index for similarity search requires at least a linear pass over the unlabeled data, this computational cost is effectively amortized over many selection rounds or other applications. As a result, our SEALS approach enables selection to scale with the size of the labeled data rather than the size of the unlabeled data, making active learning and search tractable on datasets with billions of unlabeled examples.

We empirically evaluated SEALS for both active learning and search on three large scale computer vision datasets: ImageNet (Russakovsky et al., 2015), OpenImages (Kuznetsova et al., 2020), and a proprietary dataset of 10 billion images from a large internet company. We selected 611 concepts spread across these datasets that range in prevalence from 0.203% to 0.002% (1 in 50,000) of the training examples. We evaluated three selection strategies for each concept: max entropy uncertainty sampling (Lewis & Gale, 1994), information density (Settles & Craven, 2008), and most-likely positive (Warmuth et al., 2002; 2003; Jiang et al., 2018). Across datasets, selection strategies, and concepts, SEALS achieved similar model quality and nearly the same recall of the positive examples as the baseline approaches, while improving the computational complexity by orders of magnitude. On ImageNet with a budget of 2,000 binary labels per concept (~0.31% of the unlabeled data), all baseline and SEALS approaches were within 0.011 mAP of full supervision and recalled over 50% of the positive examples. On OpenImages, SEALS reduced the candidate pool to 1% of the unlabeled data on average while remaining within 0.013 mAP and 0.1% recall of the baseline approaches. On the proprietary dataset with 10 billion images, SEALS needed an even smaller fraction of the data, about 0.1%, to match the baseline, which allowed SEALS to run on a single machine rather than a cluster. To the best of our knowledge, no other works have performed active learning at this scale. We also applied SEALS to the NLP spoiler detection dataset Goodreads (Wan et al., 2019), where it achieved the same recall as the baseline approaches while only considering less than 1% of the unlabeled data. Together, these results demonstrate that SEALS' improvements to computational efficiency make active learning and search tractable for even billion-scale datasets.

2 Related work

Active learning's iterative retraining combined with the high computational complexity of deep learning models has led to significant work on computational efficiency (Sener & Savarese, 2018; Kirsch et al., 2019; Pinsler et al., 2019; Coleman et al., 2020; Yoo & Kweon, 2019; Mayer & Timofte, 2020; Zhu & Bento, 2017). One branch of recent work has focused on selecting large batches of data to minimize the amount of retraining and reduce the number of selection rounds necessary to reach a target budget (Sener & Savarese, 2018; Kirsch et al., 2019; Pinsler et al., 2019). These

approaches introduce novel techniques to avoid selecting highly similar or redundant examples and ensure the batches are both informative and diverse. In comparison, our work aims to reduce the number of examples considered in each selection round and complements existing work on batch active learning. Many of these approaches sacrifice computational complexity to ensure diversity, and their selection methods can scale quadratically with the size of the unlabeled data. Combined with our method, these selection methods scale with the size of the labeled data rather than the unlabeled data. Outside of batch active learning, other work has tried to improve computational efficiency by either using much smaller models as cheap proxies during selection (Yoo & Kweon, 2019; Coleman et al., 2020) or by generating examples (Mayer & Timofte, 2020; Zhu & Bento, 2017). Using a smaller model reduces the amount of computation per example, but unlike our approach, it still requires making multiple passes over the entire unlabeled pool of examples. The generative approaches (Mayer & Timofte, 2020; Zhu & Bento, 2017), however, enable sub-linear runtime complexity like our approach. Unfortunately, they struggle to match the label-efficiency of traditional approaches because the quality of the generated examples is highly variable.

Active search is a sub-area of active learning that focuses on highly skewed class distributions (Garnett et al., 2012; Jiang et al., 2017; 2018; 2019). Rather than optimizing for model quality, active search aims to find as many examples from the minority class as possible. Prior work has focused on applications such as drug discovery, where the dataset sizes are limited, and labeling costs are exceptionally high. Our work similarly focuses on skewed distributions. However, we consider novel active search settings in image and text where the available unlabeled datasets are much larger, and computational efficiency is a significant bottleneck.

k **nearest neighbor (k-NN)** classifiers are popular models in active learning and search because they do not require an explicit training phase (Joshi et al., 2012; Wei et al., 2015; Garnett et al., 2012; Jiang et al., 2017; 2018). The prediction and score for each unlabeled example can be updated immediately after each new batch of labels. In comparison, our SEALS approach uses k-NN algorithms for similarity search to create and expand the candidate pool and not as a classifier. This is an important but subtle difference. While prior work avoids expensive training by using k-NN classifiers, these approaches still require evaluating all of the unlabeled examples, which can still be prohibitively expensive on large-scale datasets like the ones we consider here. SEALS targets the selection phase rather than training, presenting a novel and complementary approach.

3 METHODS

In this section, we outline the problems of active learning (Section 3.1) and search (Section 3.2) formally as well as the selection methods we accelerate using SEALS. For both, we examine the pool-based setting, where all of the unlabeled data is available at once, and examples are selected in batches to improve computational efficiency, as mentioned above. Then in Section 3.3, we describe our SEALS approach and how it further improves computational efficiency in both settings.

3.1 ACTIVE LEARNING

Pool-based active learning is an iterative process that begins with a large pool of unlabeled data $U = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Each example is sampled from the space $\mathcal{X}$ with an unknown label from the label space $\mathcal{Y} = \{1, \ldots, C\}$ as $(\mathbf{x}_i, y_i)$. We additionally assume a feature extraction function G_z to embed each $\mathbf{x}_i$ as a latent variable $G_z(\mathbf{x}_i) = \mathbf{z}_i$ and that the C concepts are unequally distributed. Specifically, there are one or more valuable rare concepts $R \subset C$ that appear in less than 1% of the unlabeled data. For simplicity, we frame this as $|R|$ binary classification problems solved independently rather than 1 multi-class classification problem with $|R|$ concepts. Initially, each rare concept has a small number of positive examples and several negative examples that serve as a labeled seed set L_r^0. The goal of active learning is to take this seed set and select up to a budget of T examples to label that produce a model A_r^T that achieves low error. For each round t in pool-based active learning, the most informative examples are selected according to the selection strategy ϕ from a pool of candidate examples $\mathcal{P}_r$ in batches of size b and labeled, as shown in Algorithm 1.

For the baseline approach, $\mathcal{P}_r = \{G_z(\mathbf{x}) \mid \mathbf{x} \in U\}$, meaning that all the unlabeled examples are considered to find the global optimal according to ϕ. Between each round, the model A_r^t is trained on all of the labeled data L_r^t, allowing the selection process to adapt.

In this paper, we considered **max entropy (MaxEnt)** uncertainty sampling (Lewis & Gale, 1994):

$$\phi_{\text{MaxEnt}}(\mathbf{z}) = -\sum_{\hat{y}} P(\hat{y}|\mathbf{z}; A_r) \log P(\hat{y}|\mathbf{z}; A_r)$$

and **information density (ID)** (Settles & Craven, 2008):

$$\phi_{\text{ID}}(\mathbf{z}) = \phi_{\text{MaxEnt}}(\mathbf{z}) \times \left(\frac{1}{|\mathcal{P}_r|} \sum_{\mathbf{z}_p \in \mathcal{P}_r} \text{sim}(\mathbf{z}, \mathbf{z}_p) \right)^{\beta}$$

where $\text{sim}(\mathbf{z}, \mathbf{z}_p)$ is the cosine similarity of the embedded examples and $\beta = 1$. Note that for binary classification, max entropy is equivalent to least confidence and margin sampling, which are also popular criteria for uncertainty sampling (Settles, 2009). While max entropy uncertainty sampling only requires a linear pass over the unlabeled data, ID scales quadratically with $|U|$ because it weights each example's informativeness by its similarity to all other examples. To improve computational performance, the average similarity score for each example can be cached after the first selection round, so subsequent rounds scale linearly. This optimization only works when G_z is fixed and would not apply to dynamic similarity calculations like those in Sener & Savarese (2018).

We explored the greedy k-centers approach from Sener & Savarese (2018) but found that it never outperformed random sampling for our experimental setup. Unlike MaxEnt and ID, k-centers does not consider the predicted labels. It tries to achieve high coverage over the entire candidate pool, of which rare concepts make up a small fraction by definition, making it ineffective for our setting.

Algorithm 1 BASELINE APPROACH

Input: unlabeled data U, labeled seed set L_r^0, feature extractor G_z, selection strategy $\phi(\cdot)$, batch size b, labeling budget T

$\mathcal{L}_r = \{(G_z(\mathbf{x}), y) \mid (\mathbf{x}, y) \in L_r^0\}$
$\mathcal{P}_r = \{G_z(\mathbf{x}) \mid \mathbf{x} \in U \text{ and } (\mathbf{x}, \cdot) \notin L_r^0\}$
repeat
  $A_r = \text{train}(\mathcal{L}_r)$
  for 1 to b **do**
    $\mathbf{z}^* = \arg\max_{\mathbf{z} \in \mathcal{P}_r} \phi(\mathbf{z})$
    $\mathcal{L}_r = \mathcal{L}_r \cup \{(\mathbf{z}^*, \text{label}(\mathbf{x}^*))\}$
    $\mathcal{P}_r = \mathcal{P}_r \setminus \{\mathbf{z}^*\}$
  end for
until $|\mathcal{L}_r| = T$

Algorithm 2 SEALS APPROACH

Input: unlabeled data U, labeled seed set L_r^0, feature extractor G_z, selection strategy $\phi(\cdot)$, batch size b, labeling budget T, k-nearest neighbors implementation $\mathcal{N}(\cdot, \cdot)$

$\mathcal{L}_r = \{(G_z(\mathbf{x}), y) \mid (\mathbf{x}, y) \in L_r^0\}$
$\mathcal{P}_r = \cup_{(\mathbf{z}, y) \in \mathcal{L}_r} \mathcal{N}(\mathbf{z}, k)$
repeat
  $A_r = \text{train}(\mathcal{L}_r)$
  for 1 to b **do**
    $\mathbf{z}^* = \arg\max_{\mathbf{z} \in \mathcal{P}_r} \phi(\mathbf{z})$
    $\mathcal{L}_r = \mathcal{L}_r \cup \{(\mathbf{z}^*, \text{label}(\mathbf{x}^*))\}$
    $\mathcal{P}_r = (\mathcal{P}_r \setminus \{\mathbf{z}^*\}) \cup \mathcal{N}(\mathbf{z}^*, k)$
  end for
until $|\mathcal{L}_r| = T$

3.2 ACTIVE SEARCH

Active search is closely related to active learning, so much of the formalism from Section 3.1 carries over. The critical difference is that rather than selecting examples to label that minimize error, the goal of active search is to maximize the number of examples from the target concept r, expressed with the natural utility function $u(L_r) = \sum_{(\mathbf{x}, y) \in L_r} \mathbb{1}\{y = r\}$). As a result, different selection strategies are favored, but the overall algorithm is the same as Algorithm 1.

In this paper, we consider an additional selection strategy to target the active search setting, **most-likely positive (MLP)** (Warmuth et al., 2002; 2003; Jiang et al., 2018):

$$\phi_{\text{MLP}}(\mathbf{z}) = P(r|\mathbf{z}; A_r)$$

Because active learning and search are similar, we evaluate all the selection criteria from Sections 3.1 and 3.2 in terms of both the error the model achieves and the number of positive examples.

3.3 SIMILARITY SEARCH FOR EFFICIENT ACTIVE LEARNING AND SEARCH (SEALS)

In this work, we propose SEALS to accelerate the inner loop of active learning and search by restricting the candidate pool of unlabeled examples. To apply SEALS, we use an efficient method for similarity search of the embedded examples (Charikar, 2002; Johnson et al., 2017) and make two modifications to the baseline approach, as shown in Algorithm 2:

1. The candidate pool $\mathcal{P}_r$ is restricted to the nearest neighbors of the labeled examples.
2. After every example is selected, we find its k nearest neighbors and update $\mathcal{P}_r$.

Computational savings. Restricting the candidate pool $\mathcal{P}_r$ to the k nearest neighbors of the labeled examples means we only apply the selection strategy to at most $k|L_r|$ examples. This can be done transparently for many selection strategies making it applicable to a wide range of active learning and search methods, even beyond the ones considered here. Finding the k nearest neighbors for each newly labeled example adds overhead, but this can be calculated efficiently with sublinear retrieval times (Charikar, 2002) and sub-second latency on billion-scale datasets (Johnson et al., 2017) for approximate approaches. As a result, the computational complexity of each selection round scales with the size of the labeled dataset rather than the unlabeled dataset. Excluding the retrieval times for the k nearest neighbors, the computational savings from SEALS are directly proportional to the pool size reduction for ϕ_{MaxEnt} and ϕ_{MLP}, which is lower bounded by $|U|/k|L_r|$. For ϕ_{ID}, the average similarity score for each example only needs to be computed once when the example is first selected. This caching means the first round scales quadratically with $|U|$ and subsequent rounds scale linearly for the baseline approach. With SEALS, each selection round scales according to $O((1 + bk)|\mathcal{P}_r|)$ because the similarity scores are calculated as examples are selected rather than all at once. The resulting computational savings of SEALS varies with the labeling budget T as the upfront cost of the baseline amortizes. Nevertheless, for large-scale datasets with millions or billions of examples, performing that first quadratic round for the baseline is prohibitively expensive.

Index construction. Generating the embeddings and indexing the data can be expensive and slow, requiring at least a linear pass over the unlabeled data. However, this cost is effectively amortized over many selection rounds, concepts, or other applications. Similarity search is a critical workload for information retrieval and powers many applications, including recommendation. Increasingly, embeddings from deep learning models are being used (Babenko et al., 2014; Babenko & Lempitsky, 2016; Johnson et al., 2017). As a result, the embeddings and index can be generated once using a generic model trained in a weak-supervision or self-supervision fashion and reused, making our approach just one of many applications using the index. Alternatively, if the data has already been passed through a predictive system (for example, to tag or classify uploaded images), the embedding could be captured and indexed at inference to avoid additional costs.

4 RESULTS

We applied SEALS to three selection strategies and performed active learning and search on three datasets: ImageNet (Russakovsky et al., 2015), OpenImages (Kuznetsova et al., 2020), and a proprietary dataset of 10 billion images. Section 4.1 details the experimental setup for each dataset and the inputs used for both the baseline approach (Algorithm 1) and our proposed method, SEALS (Algorithm 2). Sections 4.2 and 4.3 present the empirical results for active learning and search. Section 4.4 explores the structure of the concepts through the nearest neighbor graphs and embeddings.

Across selection strategies, datasets, and concepts, SEALS using ResNet-50 (He et al., 2016) embeddings performed similarly to the baseline while only considering a fraction of the unlabeled data U in the candidate pool for each concept $\mathcal{P}_r$. For MLP and MaxEnt, the smaller candidate pool from SEALS sped-up the selection runtime by over $180\times$ on OpenImages. This allowed us to run active learning and search efficiently on an industrial scale dataset with 10 billion images. The improvements were even larger for information density. On ImageNet, SEALS dropped the time for the first selection round from over 75 minutes to 1.5 seconds, over a $3000\times$ improvement. On OpenImages, the baseline for information density ran for over 24 hours without completing a single round, while SEALS took less than 3 minutes to perform 19 rounds. We observed similar results with self-supervised embeddings using SimCLR (Chen et al., 2020) for ImageNet (Appendix A.3) and using SentenceBERT (Reimers & Gurevych, 2019) for Goodreads spoiler detection (Appendix A.8).

Table 1: Summary of datasets

	Number of Concepts ($\lvert R \rvert$)	**Embedding Model** (G_z)	**Number of Examples** ($\lvert U \rvert$)	**Fraction Positive** $\left(\frac{\sum \mathbb{1}\{y=r\}}{\lvert U \rvert}\right)$
ImageNet (Russakovsky et al., 2015)	450	ResNet-50 (He et al., 2016) (500 classes)	639,906	0.114-0.203%
OpenImages (Kuznetsova et al., 2020)	153	ResNet-50 (He et al., 2016) (1000 classes)	6,816,296	0.002-0.088%
10B images (proprietary)	8	ResNet-50 (He et al., 2016) (1000 classes)	10,094,719,767	-

4.1 EXPERIMENTAL SETUP

Across all datasets and selection strategies, we followed the same general procedure for both active learning and search. Because we are interested in rare concepts, we kept the number of initial positive examples small. We evaluated three settings, with 5, 20, and 50 positives, but only included the results with the smallest size in this section. The others are shown in Appendix A.1. For each setting, negative examples were randomly selected at a ratio of 19 negative examples to every positive example to form the seed set L_r^0. The slightly higher number of negatives in the initial seed set improved average precision on the validation set across all three datasets. The batch size b for each selection round was the same as the size of the initial seed set. For the seed set of 5 positive and 95 negative examples shown below, b was 100, and the labeling budget T was 2,000 examples.

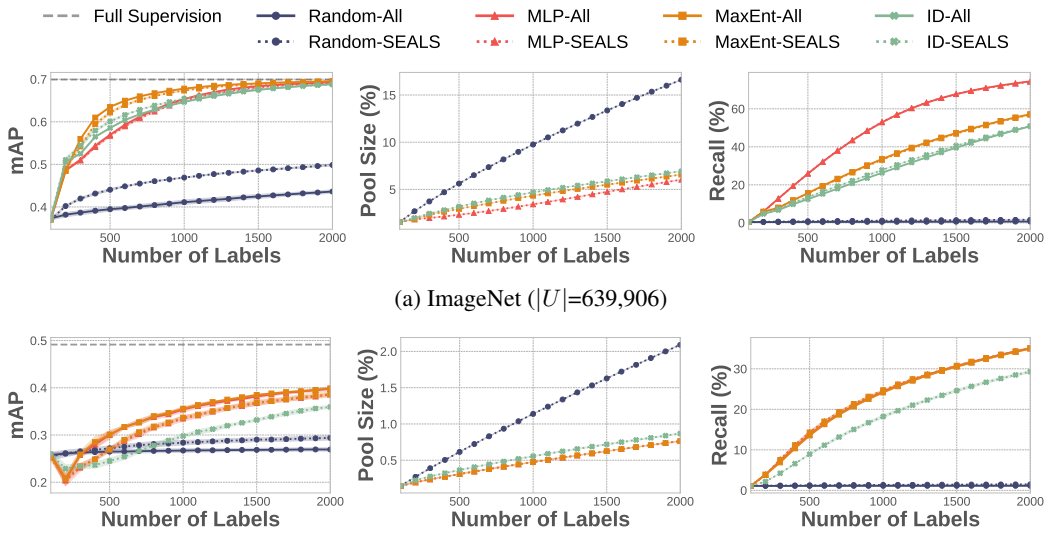
As the binary classifier for each concept A_r, we used logistic regression trained on the embedded examples. For active learning, we calculated average precision on the test data for each binary concept classifier after each selection round. For active search, we count the number of positive examples labeled so far. We take the mean average precision (mAP) and number of positives across concepts, run each experiment 5 times, and report the mean and standard deviation.

For similarity search, we used locality-sensitive hashing (LSH) (Charikar, 2002) implemented in Faiss (Johnson et al., 2017) with Euclidean distance for all datasets aside from the 10 billion images dataset. This simplified our implementation, so the index could be created quickly and independently for each concept and configuration, allowing experiments to run in parallel trivially. However, retrieval times for this approach were not as fast as Johnson et al. (2017) and made up a larger part of the overall active learning loop. In practice, the search index can be heavily optimized and tuned for the specific data distribution, leading to computational savings closer to the improvements described in Section 3.3 and differences in the "Selection" portion of the runtimes in Table 2.

We split the data, selected concepts, and created embeddings as detailed below and summarized in Table 1. Note that our approach does not constrain the choice of G_z, which allows for many network architectures. As representations continue to improve with new self-supervision, generative, or transfer learning techniques, SEALS is still applicable and performance will also likely improve.

ImageNet (Russakovsky et al., 2015) has 1.28 million training images spread over 1000 classes. To simulate rare concepts, we split the data in half, using 500 classes to train the feature extractor G_z and treating the other 500 classes as unseen concepts. For G_z, we used ResNet-50 but added a bottleneck layer before the final output to reduce the dimension of the embeddings to 256. We kept all of the other training hyperparameters the same as in He et al. (2016). We extracted features from the bottleneck layer and applied l^2 normalization. In total, the 500 unseen concepts had 639,906 training examples that served as the unlabeled pool. We used 50 concepts for validation, leaving the remaining 450 concepts for our final experiments. The number of examples for each concept varied slightly, ranging from 0.114-0.203% of $|U|$. The 50,000 validation images were used as the test set.

OpenImages (Kuznetsova et al., 2020) has 7.34 million images with human-verified labels spread over 19,958 classes, taken as an unbiased sample from Flickr. However, only 6.82 million images were still available in the training set at the time of writing. As a feature extractor, we took ResNet-50 pre-trained on all of ImageNet and used the l^2 normalized output from the bottleneck layer. As rare concepts, we randomly selected 200 classes with between 100 to 6,817 positive training examples. We reviewed the selected classes and removed 47 classes that overlapped with ImageNet. The remaining 153 classes appeared in 0.002-0.088% of the data. We used the same hyperparameters as the ImageNet experiments and the OpenImages predefined test split for evaluation.



(a) ImageNet ($|U|$=639,906)

(b) OpenImages ($|U|$=6,816,296)

Figure 1: Active learning and search on ImageNet (top) and OpenImages (bottom). Across datasets and strategies, SEALS with $k = 100$ performed similarly to the baseline approach in terms of both the error the model achieved for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the data U (middle).

10 billion (10B) images from a large internet company were used to test SEALS' scalability. For the feature extractor, we used the same pre-trained ResNet-50 model as the OpenImages experiments. We also selected 8 additional classes from OpenImages as rare concepts: rat, sushi, bowling, beach, hawk, cupcake, and crowd. This allowed us to use the predefined test split from OpenImages for evaluation. Unlike the other datasets, we hired annotators to label images as they were selected and used a proprietary index to achieve low latency retrieval times to capture a real-world setting.

4.2 Active Learning

Across datasets and selection strategies, SEALS performed similarly to the baseline approaches that considered all of the unlabeled data in the candidate pool, as shown in Figures 1 and 2.

ImageNet. With a labeling budget of 2,000 examples per concept (~0.31% of $|U|$), all baseline and SEALS approaches ($k = 100$) were within 0.011 mAP of the 0.699 mAP achieved with full supervision. In contrast, random sampling (Random-All) only achieved 0.436 mAP. MLP-All, MaxEnt-All, and ID-All achieved mAPs of 0.693, 0.695, and 0.688, respectively, while the SEALS equivalents were all within 0.001 mAP at 0.692, 0.695, and 0.688 respectively and considered less than 7% of the unlabeled data. The resulting selection runtime for MLP-SEALS and MaxEnt-SEALS dropped by over 25×, leading to a 3.6× speed-up overall (Table 2). The speed-up was even larger for ID-SEALS, ranging from about 45× at 2,00 labels to 3000× at 200 labels. Even at a per-class level, the results were highly correlated with Pearson correlation coefficients of 0.9998 or more (Figure 10a in the Appendix). The reduced skew from the nearest neighbor expansion of the initial seed set only accounted for a small part of the improvement, as Random-SEALS achieved an mAP of 0.498.

OpenImages. The gap between the baseline approaches and SEALS widened slightly for OpenImages. At 2,000 labels per concept (~0.029% of $|U|$), MaxEnt-All and MLP-All achieved 0.399 and 0.398 mAP, respectively, while MaxEnt-SEALS and MLP-SEALS both achieved 0.386 mAP and considered less than 1% of the data. This sped-up the selection time by over 180× and the total time by over 3×. Increasing k to 1,000 significantly narrowed this gap for MaxEnt-SEALS and MLP-SEALS, improving mAP to 0.395, as shown in the Appendix (Figure 7). Moreover, SEALS made ID tractable on OpenImages by reducing the candidate pool to 1% of the unlabeled data, whereas ID-All ran for over 24 hours in wall-clock time without completing a single round (Table 2).

Table 2: Wall clock runtimes for varying selection strategies on ImageNet and OpenImages. The last 3 columns break the total time down into 1) the time to apply the selection strategy to the candidate pool, 2) the time to find the k nearest neighbors (k-NN) for the newly labeled examples, and 3) the time to train logistic regression on the currently labeled examples. Despite using a simple LSH index for similarity search, SEALS substantially improved runtimes across datasets and strategies.

Dataset	Budget T	Strategy ϕ	mAP/AUC	Recall (%)	Pool Size (%)	Total Time (seconds)	Time Breakdown (seconds)		
							Selection	k-NN	Training
ImageNet	2,000	MaxEnt-All	0.695	57.2	100.0	45.23	44.65	-	0.59
		MaxEnt-SEALS	0.695	56.9	6.6	12.49	1.73	10.27	0.50
		MLP-All	0.693	74.5	100.0	43.32	42.75	-	0.57
		MLP-SEALS	0.692	74.2	6.0	12.03	1.48	9.94	0.63
		ID-All	0.688	50.8	100.0	4654.59	4653.55	-	1.05
		ID-SEALS	0.688	50.9	6.9	104.57	94.22	9.76	0.60
	1,000	ID-All	0.646	26.3	100.0	4620.04	4619.78	-	0.28
		ID-SEALS	0.654	27.8	4.7	36.66	31.95	4.56	0.17
	500	ID-All	0.586	12.5	100.0	4602.64	4602.57	-	0.09
		ID-SEALS	0.601	13.5	3.2	9.75	7.75	1.95	0.05
	200	ID-All	0.506	4.7	100.0	4588.76	4588.73	-	0.04
		ID-SEALS	0.511	4.8	2.0	1.53	1.03	0.49	0.02
OpenImages	2,000	MaxEnt-All	0.399	35.0	100.0	295.20	294.78	-	0.42
		MaxEnt-SEALS	0.386	35.1	0.8	80.61	1.56	78.63	0.43
		MLP-All	0.398	35.1	100.0	285.27	284.88	-	0.40
		MLP-SEALS	0.386	35.1	0.8	82.18	1.48	80.27	0.44
		ID-All	-	-	100.0	>24 hours	>24 hours	-	-
		ID-SEALS	0.359	29.3	0.9	129.79	48.98	80.40	0.41

10B images. Despite the unprecedented scale and limited pool size, SEALS performed similarly to the baseline approaches that scanned all 10 billion images. At a budget of 1,500 labels, MaxEnt-SEALS (k=10K) achieved a similar mAP to the baseline (0.504 vs. 0.508 mAP), while considering only about 0.1% of the data. This reduction allowed MaxEnt-SEALS to finish selection rounds in seconds on a single 24-core machine, while MaxEnt-All took minutes on a cluster with tens of thousands of cores. MLP-SEALS performed poorly at this scale because, for any image, there are likely many redundant or near-duplicate examples that provide little additional value.

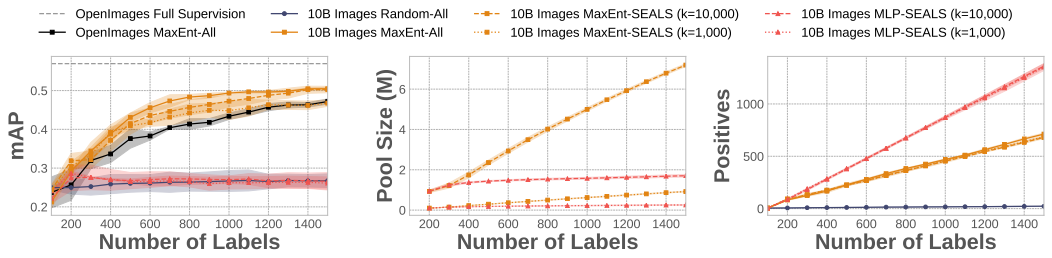
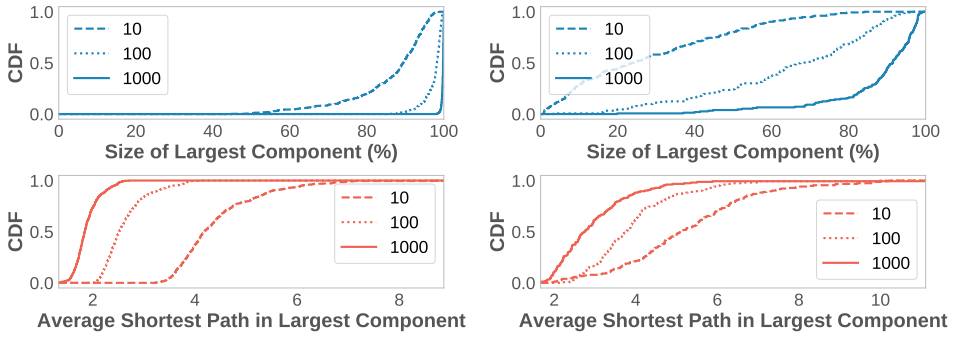


Figure 2: Active learning and search on a proprietary dataset of 10 billion images. Across strategies, SEALS with $k = 10,000$ performed similarly to the baseline approach in terms of both the error the model achieved for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the data U (middle).

4.3 ACTIVE SEARCH

As shown in Figures 1 and 2, SEALS recalled nearly the same number of positive examples as the baseline approaches did for all of the considered concepts, datasets, and selection strategies.

ImageNet. Unsurprisingly, MLP-All and MLP-SEALS significantly outperformed all of the other selection strategies for active search. At 2,000 labeled examples per concept, both approaches recalled over 74% of the positive examples for each concept at 74.5% and 74.2% recall, respectively. MaxEnt-All and MaxEnt-SEALS had a similar gap of 0.3%, labeling 57.2% and 56.9% of positive examples, while ID-All and ID-SEALS were even closer with a gap of only 0.1% (50.8% vs. 50.9%). Nearly all of the gains in recall are due to the selection strategies rather than the reduced skew in the initial seed, as Random-SEALS increased the recall by less than 1.0% over Random-All.



(a) ImageNet (b) OpenImages

Figure 3: Measurements of the latent structure of unseen concepts in ImageNet (left) and OpenImages (right). Across datasets, the k-nearest neighbor graph of unseen concepts was well connected, forming large connected components (top) for even moderate values of k. The components were tightly packed, leading to short paths between examples (bottom).

OpenImages. The gap between the baseline approaches and SEALS was even closer on OpenImages despite considering a much smaller fraction of the overall unlabeled pool. MLP-All, MLP-SEALS, MaxEnt-SEALS, and MaxEnt-All were all within 0.1% with ˜35% recall at 2,000 labels per concept. ID-SEALS had a recall of 29.3% but scaled nearly as well as the linear approaches.

10B images. SEALS performed as well as the baseline approach despite considering less than 0.1% of the data and collected 2 orders of magnitude more positive examples than random sampling.

4.4 Latent structure of unseen concepts

To better understand why and when SEALS works, we analyzed the nearest neighbor graph across concepts and values of k. Figure 3 shows the cumulative distribution functions (CDF) for the largest connected component within each concept and the average shortest paths between examples in that component. The 10B images dataset was excluded because only a few thousand examples were labeled. The largest connected component gives a sense of how much of the concept SEALS can reach, while the average shortest path serves as a proxy for how long it will take to explore.

In general, SEALS performed better for concepts that formed larger connected components and had shorter paths between examples (Figure 11 in the Appendix). For most concepts in ImageNet, the largest connected component contained the majority of examples, and the paths between examples were very short. These tight clusters explain why so few examples were needed to learn accurate binary concept classifiers, as shown in Section 4.2, and why SEALS recovered ˜74% of positive examples on average while only labeling ˜0.31% of the data. If we constructed the candidate pool by randomly selecting examples, mAP and recall would drop for all strategies (Appendix A.5). The concepts were so rare that the randomly chosen examples were not close to the decision boundary. For OpenImages, rare concepts were more fragmented, but each component was fairly tight, leading to short paths between examples. This fragmentation partly explains the gap between SEALS and the baselines in Section 4.2, and why increasing k closed it. As k increases, so does the candidate pool and the computational complexity, creating a trade-off between computational efficiency and the labeling efficiency of the underlying selection strategies. However, even for small values of k, there were significant gains over random sampling, as shown in Figures 6 and 7 in the Appendix.

5 Conclusion

In this work, we introduced Similarity search for Efficient Active Learning and Search (SEALS) as a simple approach to accelerate active learning and search that can be applied to a wide range of existing algorithms. SEALS restricted the candidate pool for labeling to the nearest neighbors of the currently labeled set instead of scanning over all of the unlabeled data. Across three large datasets, three selection strategies, and 611 concepts, we found that SEALS achieved similar model quality and recall of positive examples while improving computational efficiency by orders of magnitude.

REFERENCES

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016.

Dario Amodei and Danny Hernandez. Ai and compute, May 2018. URL https://blog.openai.com/ai-and-compute/.

Khalid Ashmawy, Shouheng Yi, and Alex Chao. Searchable ground truth: Querying uncommon scenarios in self-driving car development. https://eng.uber.com/searchable-ground-truth-atg/, 10 2019.

Artem Babenko and Victor Lempitsky. Efficient indexing of billion-scale datasets of deep descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2055–2063, 2016.

Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pp. 584–599. Springer, 2014.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 380–388, 2002.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google, 2013. URL http://arxiv.org/abs/1312.3005.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJg2b0VYDr.

Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard Mann. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pp. 843–850, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2016.

Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2019.

Shali Jiang, Gustavo Malkomes, Geoff Converse, Alyssa Shofner, Benjamin Moseley, and Roman Garnett. Efficient nonmyopic active search. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1714–1723, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/jiang17d.html.

Shali Jiang, Gustavo Malkomes, Matthew Abbott, Benjamin Moseley, and Roman Garnett. Efficient nonmyopic batch active search. In *Advances in Neural Information Processing Systems*, pp. 1099–1109, 2018.

Shali Jiang, Roman Garnett, and Benjamin Moseley. Cost effective active search. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 4880–4889. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8734-cost-effective-active-search.pdf.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Coverage optimized active learning for k-nn classifiers. In *2012 IEEE International Conference on Robotics and Automation*, pp. 5353–5358. IEEE, 2012.

Andrej Karpathy. Train ai 2018 - building the software 2.0 stack, 2018. URL https://vimeo.com/272696002.

Andrej Karpathy. Ai for full-self driving, 2020. URL https://youtu.be/hx7BXih7zx8.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pp. 7024–7035, 2019.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

Kevin Lee, Vijay Rao, and William Christie Arnold. Accelerating facebook's infrastructure with application-specific hardware. https://engineering.fb.com/data-center-engineering/accelerating-infrastructure/, 3 2019.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 3–12. Springer-Verlag New York, Inc., 1994.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 3071–3079, 2020.

Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*, pp. 6356–6367, 2019.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Burr Settles. From theories to queries: Active learning in practice. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov (eds.), *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pp. 1–18, Sardinia, Italy, 16 May 2011. PMLR. URL http://proceedings.mlr.press/v16/settles11a.html.

Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6 (1):1–114, 2012.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 1070–1079, USA, 2008. Association for Computational Linguistics.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m. *Communications of the ACM*, 59(2):64–73, Jan 2016. ISSN 1557-7317. doi: 10.1145/2812802. URL http://dx.doi.org/10.1145/2812802.

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

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. Fine-grained spoiler detection from large-scale review corpora. In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 2605–2610. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1248. URL https://doi.org/10.18653/v1/p19-1248.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pp. 7029–7039, 2017.

Manfred K Warmuth, Jun Liao, Gunnar Rätsch, Michael Mathieson, Santosh Putta, and Christian Lemmen. Active learning with support vector machines in the drug discovery process. *Journal of chemical information and computer sciences*, 43(2):667–673, 2003.

Manfred KK Warmuth, Gunnar Rätsch, Michael Mathieson, Jun Liao, and Christian Lemmen. Active learning in the drug discovery process. In *Advances in Neural information processing systems*, pp. 1449–1456, 2002.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pp. 1954–1963, 2015.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 93–102, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pp. 649–657, 2015.

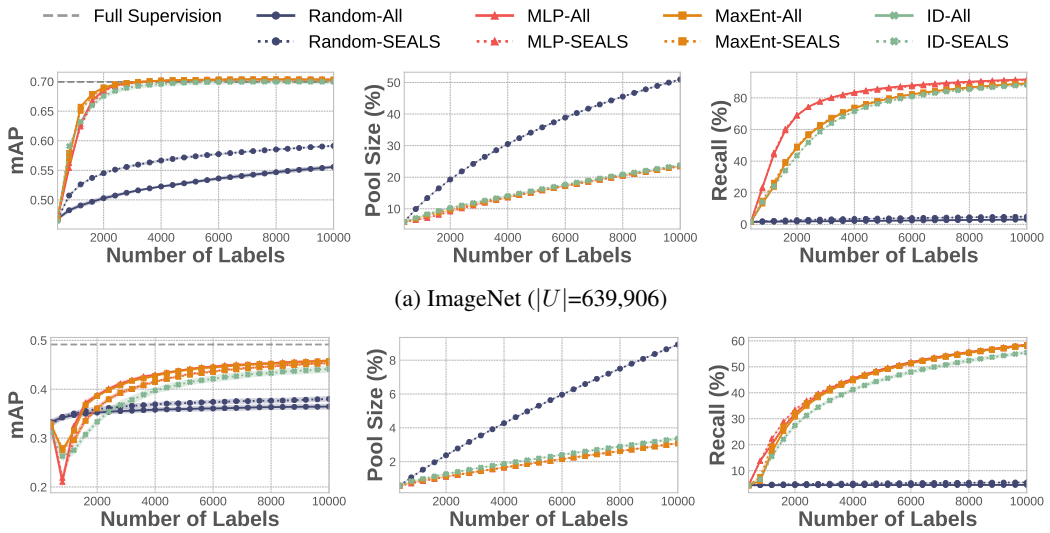
Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5409–5418, 2017.

Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pp. 19–27, 2015.

A APPENDIX

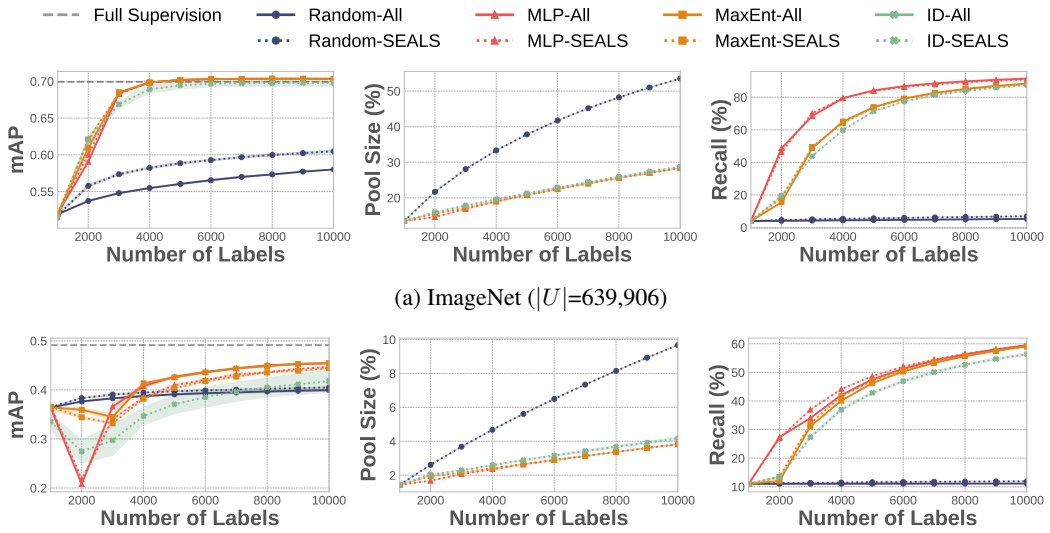
A.1 NUMBER OF INITIAL POSITIVES



(a) ImageNet ($|U|$=639,906)

(b) OpenImages ($|U|$=6,816,296)

Figure 4: **Active learning and search with 20 positive seed examples** and a labeling budget of 10,000 examples on ImageNet (top) and OpenImages (bottom). Across datasets and strategies, SEALS with $k = 100$ performs similarly to the baseline approach in terms of both the error the model achieves for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the unlabeled data U (middle).



(a) ImageNet ($|U|$=639,906)

(b) OpenImages ($|U|$=6,816,296)

Figure 5: **Active learning and search with 50 positive seed examples** and a labeling budget of 10,000 examples on ImageNet (top) and OpenImages (bottom). Across datasets and strategies, SEALS with $k = 100$ performs similarly to the baseline approach in terms of both the error the model achieves for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the unlabeled data U (middle).

A.2 IMPACT OF k ON SEALS

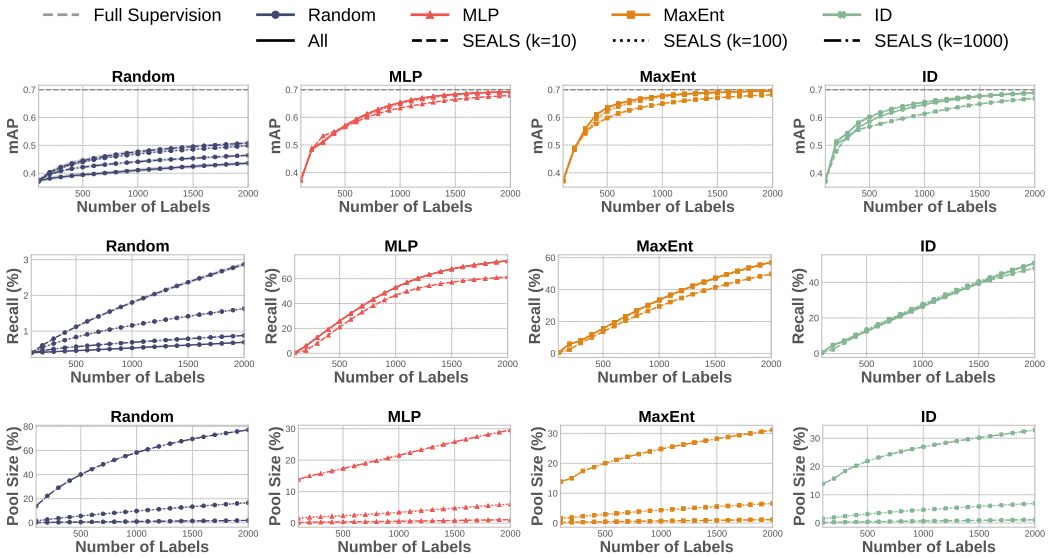


Figure 6: **Impact of increasing k on ImageNet ($|U|$=639,906).** Larger values of k help to close the gap between SEALS and the baseline approach that considers all of the unlabeled data for both active learning (top) and active search (middle). However, increasing k also increases the candidate pool size (bottom), presenting a trade-off between labeling efficiency and computational efficiency.

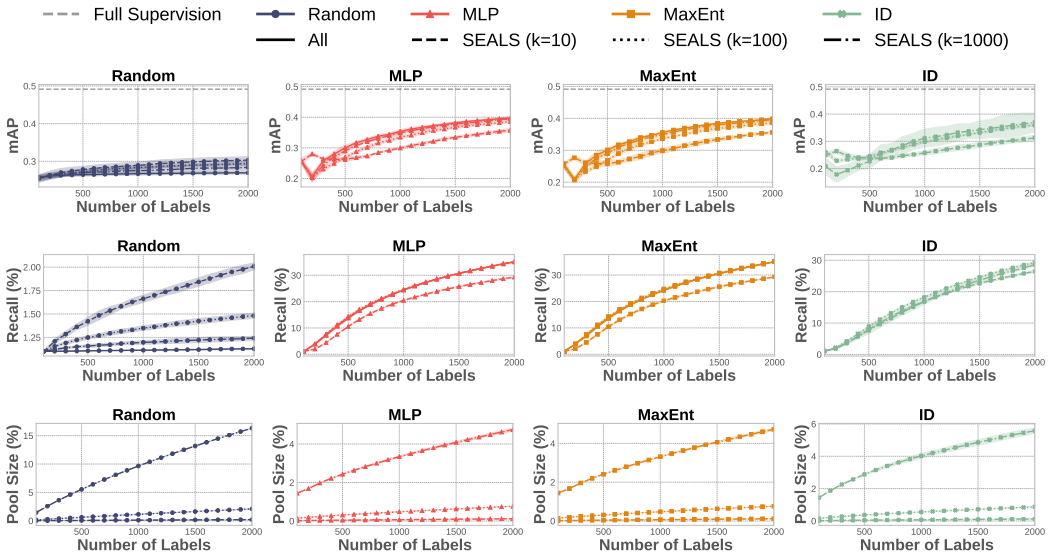


Figure 7: **Impact of increasing k on OpenImages ($|U|$=6,816,296).** Larger values of k help to close the gap between SEALS and the baseline approach that considers all of the unlabeled data for both active learning (top) and active search (middle). However, increasing k also increases the candidate pool size (bottom), presenting a trade-off between labeling efficiency and computational efficiency.

A.3 SELF-SUPERVISED EMBEDDINGS (SIMCLR) ON IMAGENET

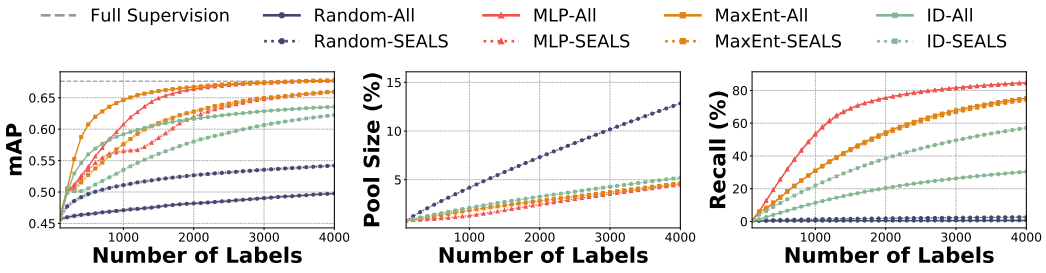


Figure 8: Active learning and search on ImageNet with self-supervised embeddings from SimCLR (Chen et al., 2020). Because the self-supervised training for the embeddings did not use the labels, results are average across all 1,000 classes and $|U|$=1,281,167. To compensate for the larger unlabeled pool, we extended the total labeling budget to 4,000 compared to the 2,000 used in Figure 1. Across strategies, SEALS with $k = 100$ substantially outperforms random sampling in terms of both the mAP the model achieves for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the data U (middle). For active learning, the gap between the baseline and SEALS approaches is slightly larger than in Figure 1, which is likely due to the larger pool size and increased average shortest paths (see Figure 9).

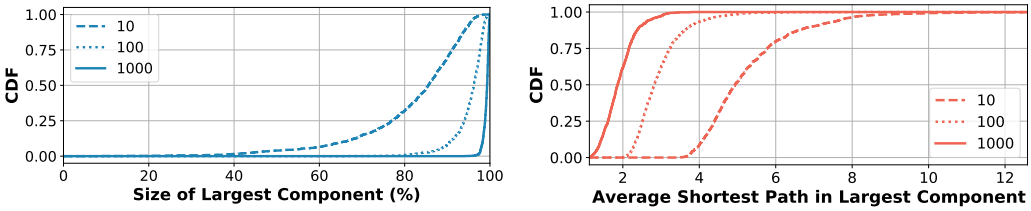
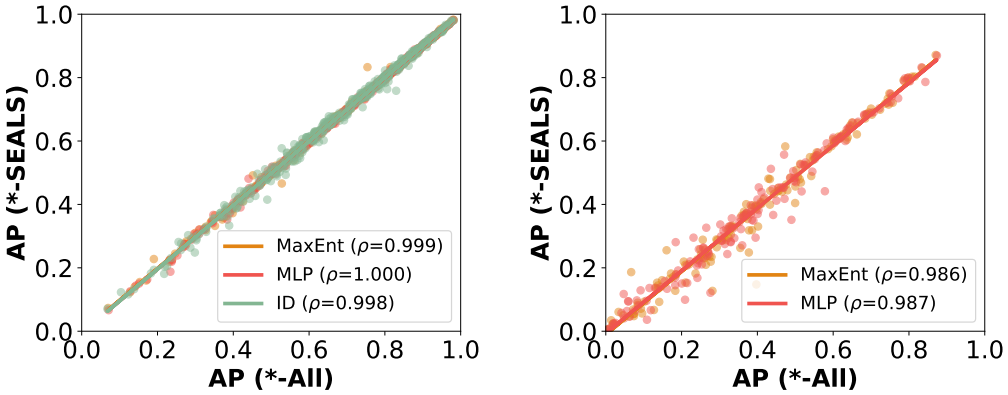


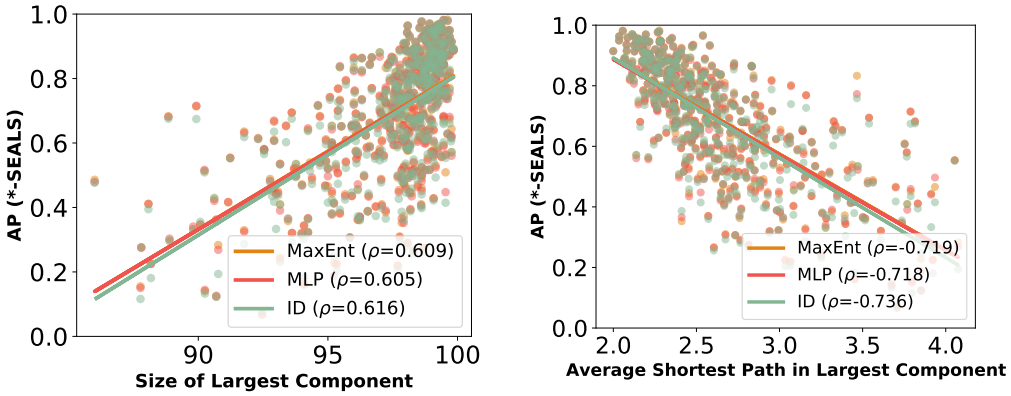
Figure 9: Measurements of the latent structure of unseen concepts in ImageNet with self-supervised embeddings from SimCLR (Chen et al., 2020). In comparison to Figure 3a, the k-nearest neighbor graph for unseen concepts was still well connected, forming large connected components (left) for even moderate values of k, but the average shortest path between examples was slightly longer (right). The increased path length is not too surprising considering the fully supervised model still outperformed the linear evaluation of the self-supervised embeddings in Chen et al. (2020).

A.4 PER CLASS AP CORRELATION

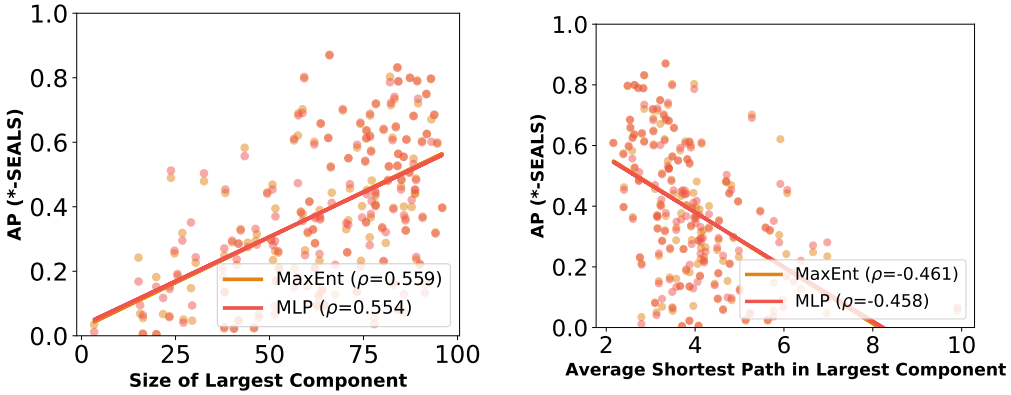


(a) ImageNet (b) OpenImages

Figure 10: The per-class APs of SEALS were highly correlated to the baseline approaches (*-All) for active learning on ImageNet (right) and OpenImages (left). On OpenImages with $k = 100$ and a budget of 2,000 labels, the Pearson's correlation (ρ) between the baseline and SEALS for the average precision of individual classes was 0.986 for MaxEnt and 0.987 for MLP. The least-squares fit had a slope of 0.99 and y-intercept of -0.01. On ImageNet, the correlations were even higher.



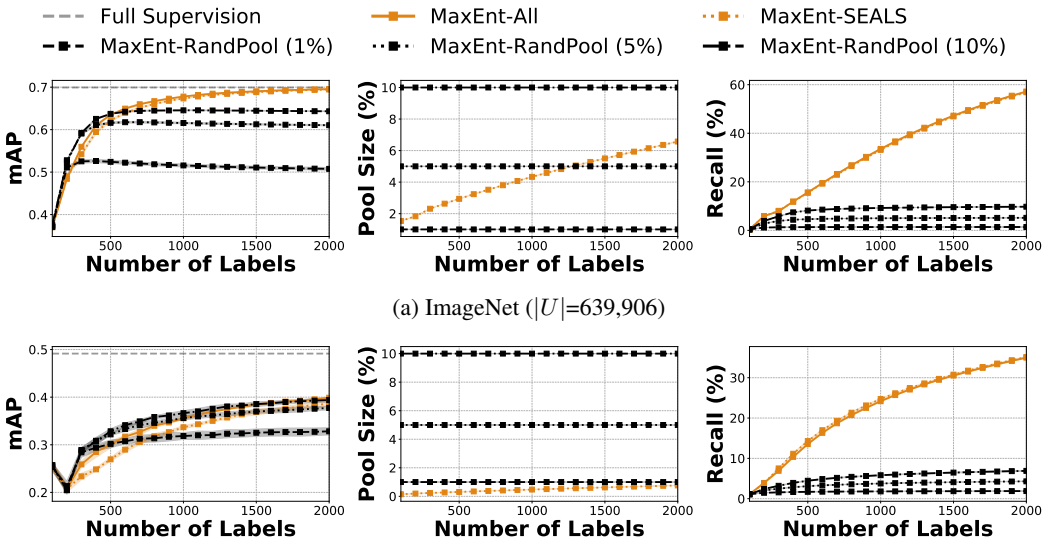
(a) ImageNet



(b) OpenImages

Figure 11: SEALS achieved higher APs for classes that formed larger connected components (left) and had shorter paths between examples (right) in ImageNet (top) and OpenImages (bottom).

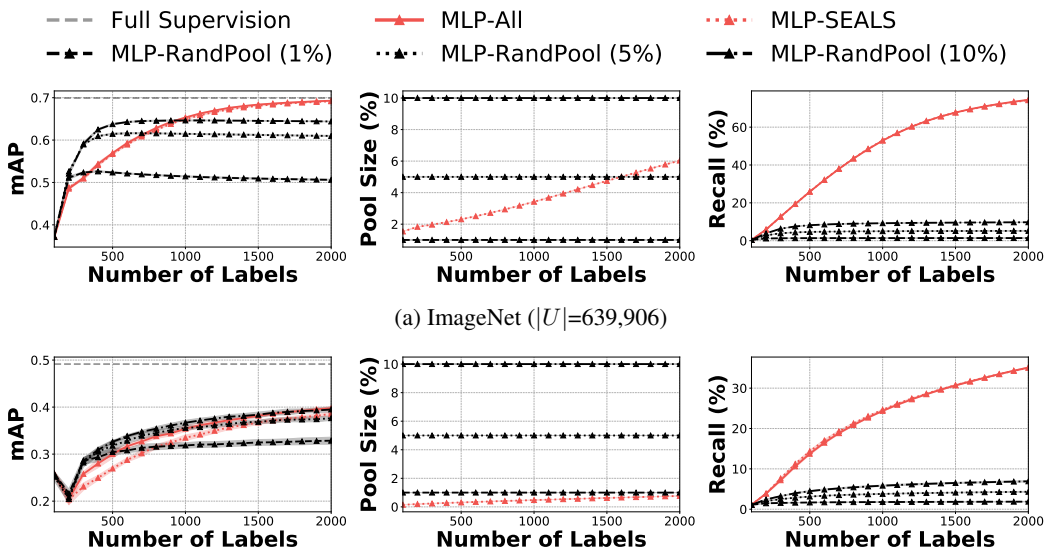
A.5 COMPARISON TO POOL OF RANDOMLY SELECTED EXAMPLES



(a) ImageNet ($|U|$=639,906)

(b) OpenImages ($|U|$=6,816,296)

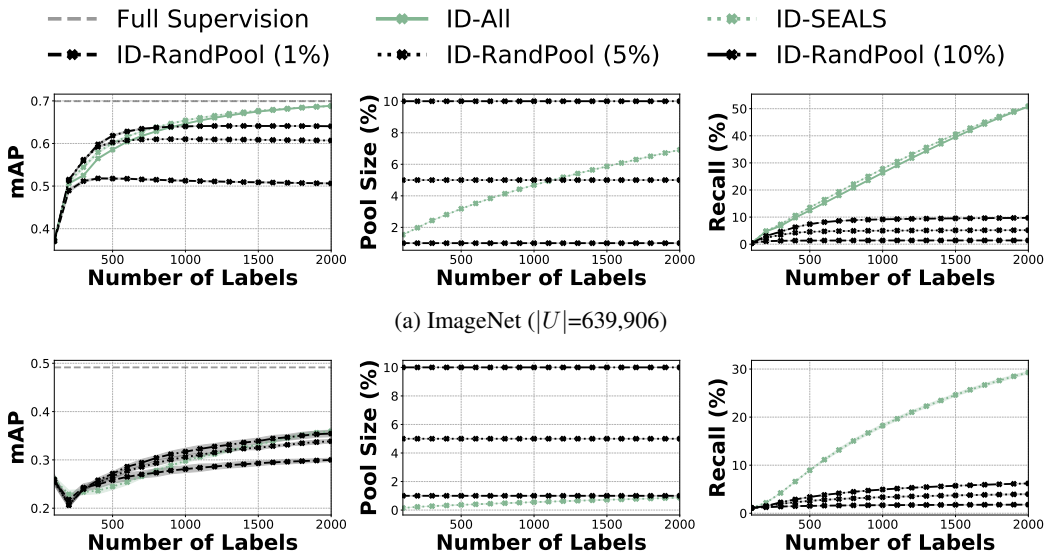
Figure 12: MaxEnt-SEALS ($k = 100$) versus MaxEnt applied to a candidate pool of randomly selected examples (RandPool). Because the concepts we considered were so rare, as is often the case in practice, randomly chosen examples are unlikely to be close to the decision boundary, and a much larger pool is required to match SEALS. On ImageNet (top), MaxEnt-SEALS outperformed MaxEnt-RandPool in terms of both the error the model achieves for active learning (left) and the recall of positive examples for active search (right) even with a pool containing 10% of the data (middle). On Openimages (bottom), MaxEnt-RandPool needed at least $5\times$ as much data to match MaxEnt-SEALS for active learning and failed to achieve similar recall even with $10\times$ the data.



(a) ImageNet ($|U|$=639,906)

(b) OpenImages ($|U|$=6,816,296)

Figure 13: MLP-SEALS ($k = 100$) versus MLP applied to a candidate pool of randomly selected examples (RandPool). Because the concepts we considered were so rare, as is often the case in practice, randomly chosen examples are unlikely to be close to the decision boundary, and a much larger pool is required to match SEALS. On ImageNet (top), MLP-SEALS outperformed MLP-RandPool in terms of both the error the model achieves for active learning (left) and the recall of positive examples for active search (right) even with a pool containing 10% of the data (middle). On Openimages (bottom), MLP-RandPool needed at least $5\times$ as much data to match MLP-SEALS for active learning and failed to achieve similar recall even with $10\times$ the data.



(a) ImageNet ($|U|$=639,906)

(b) OpenImages ($|U|$=6,816,296)

Figure 14: ID-SEALS ($k = 100$) versus ID applied to a candidate pool of randomly selected examples (RandPool). Because the concepts we considered were so rare, as is often the case in practice, randomly chosen examples are unlikely to be close to the decision boundary, and a much larger pool is required to match SEALS. On ImageNet (top), ID-SEALS outperformed ID-RandPool in terms of both the error the model achieves for active learning (left) and the recall of positive examples for active search (right) even with a pool containing 10% of the data (middle). On Openimages (bottom), ID-RandPool needed at least $5\times$ as much data to match ID-SEALS for active learning and failed to achieve similar recall even with $10\times$ the data.

A.6 ACTIVE LEARNING ON EACH SELECTED CLASS FROM OPENIMAGES

Table 3: **Top $\frac{1}{3}$ of classes from Openimages for active learning.** (1 of 3) Average precision for each selected class (153 total) from OpenImages with a labeling budget of 2,000 examples. Classes are in descending order based on the performance gain of MaxEnt-SEALS over Random-All.

Label Name	Display Name	Total Positives	Random (All)	MaxEnt (SEALS)	MaxEnt (All)	Full Supervision
/m/018gpn	Hurdling	269	0.26 ± 0.094	0.80 ± 0.005	0.79 ± 0.005	0.80
/m/0dqb5	Citrus	796	0.34 ± 0.191	0.87 ± 0.001	0.87 ± 0.003	0.87
/m/0btmb	Superhero	968	0.17 ± 0.106	0.70 ± 0.008	0.70 ± 0.021	0.67
/m/09wzfj	California roll	368	0.05 ± 0.018	0.56 ± 0.007	0.56 ± 0.007	0.58
/m/01xc8d	Rope	618	0.29 ± 0.243	0.80 ± 0.003	0.81 ± 0.003	0.74
/m/0ll1f78	Shrimp	907	0.07 ± 0.024	0.56 ± 0.002	0.56 ± 0.009	0.58
/m/09l0j_	Ring	676	0.15 ± 0.029	0.61 ± 0.012	0.64 ± 0.018	0.64
/m/0dwxr	Modern pentathlon	772	0.13 ± 0.085	0.58 ± 0.010	0.47 ± 0.257	0.51
/m/02yhp1	San Pedro cactus	318	0.17 ± 0.114	0.62 ± 0.003	0.63 ± 0.005	0.71
/m/0bpn3c2	Skateboarding Equipment	862	0.20 ± 0.105	0.62 ± 0.073	0.66 ± 0.013	0.66
/m/0cyz6w	Whole food	708	0.18 ± 0.115	0.58 ± 0.020	0.60 ± 0.020	0.57
/m/01943w	Ibis	259	0.29 ± 0.068	0.68 ± 0.010	0.69 ± 0.009	0.66
/m/03d8m_	Monster truck	286	0.41 ± 0.124	0.79 ± 0.010	0.80 ± 0.004	0.81
/m/01s0ps	Electric piano	345	0.24 ± 0.106	0.61 ± 0.024	0.60 ± 0.020	0.48
/m/01cz4w	Bilberry	228	0.10 ± 0.091	0.45 ± 0.031	0.45 ± 0.026	0.32
/m/0j_5b	Monoplane	756	0.13 ± 0.072	0.48 ± 0.014	0.43 ± 0.174	0.48
/m/058qzx	Kitchen knife	360	0.32 ± 0.130	0.66 ± 0.003	0.65 ± 0.004	0.66
/m/0cghn	Mcdonnell douglas f/a-18 hornet	160	0.11 ± 0.065	0.44 ± 0.041	0.47 ± 0.027	0.37
/m/05z55	Pasta	954	0.42 ± 0.104	0.75 ± 0.014	0.75 ± 0.007	0.79
/m/05h5q_s	Glider	393	0.08 ± 0.045	0.40 ± 0.004	0.40 ± 0.004	0.48
/m/026545k	Elk	353	0.15 ± 0.018	0.46 ± 0.014	0.48 ± 0.005	0.45
/m/0dxb5	Berry	874	0.30 ± 0.117	0.61 ± 0.007	0.61 ± 0.007	0.69
/m/01ngsj	Daylily	391	0.20 ± 0.077	0.51 ± 0.006	0.50 ± 0.003	0.49
/m/04l3lm	Shooting range	189	0.38 ± 0.127	0.69 ± 0.011	0.69 ± 0.008	0.68
/m/0wgq98_	Skating	561	0.17 ± 0.017	0.48 ± 0.028	0.43 ± 0.147	0.40
/m/01xq0k1	Cattle	5995	0.37 ± 0.085	0.67 ± 0.007	0.68 ± 0.006	0.74
/m/0cfk35	Chocolate truffle	288	0.10 ± 0.047	0.39 ± 0.015	0.40 ± 0.020	0.42
/m/0cnxs6x	Tooth	976	0.16 ± 0.053	0.44 ± 0.069	0.48 ± 0.074	0.56
/m/01ll1r	Briefs	539	0.15 ± 0.074	0.43 ± 0.008	0.44 ± 0.005	0.46
/m/04fyb1	Sirloin steak	297	0.14 ± 0.092	0.42 ± 0.007	0.42 ± 0.032	0.46
/m/0cl6l	Straw	547	0.33 ± 0.142	0.61 ± 0.004	0.62 ± 0.011	0.61
/m/06lk1	Rat	1151	0.32 ± 0.098	0.60 ± 0.003	0.60 ± 0.003	0.61
/m/03fwl	Goat	1190	0.17 ± 0.056	0.44 ± 0.017	0.45 ± 0.019	0.61
/m/033cnk	Egg (Food)	1193	0.14 ± 0.089	0.40 ± 0.061	0.37 ± 0.019	0.63
/m/058hry	Smoothie	330	0.15 ± 0.039	0.41 ± 0.007	0.41 ± 0.004	0.38
/m/020dp	Cranberry	450	0.13 ± 0.099	0.39 ± 0.041	0.39 ± 0.012	0.37
/m/0h8n982	Shelving	810	0.27 ± 0.097	0.53 ± 0.012	0.53 ± 0.013	0.51
/m/01bmhj	Thumb	895	0.07 ± 0.133	0.32 ± 0.178	0.39 ± 0.043	0.41
/m/0cnmr	Fur	834	0.08 ± 0.087	0.33 ± 0.013	0.33 ± 0.016	0.31
/m/0g9vs81	Steamed rice	580	0.10 ± 0.055	0.35 ± 0.039	0.37 ± 0.032	0.48
/m/0176mf	Belt	467	0.06 ± 0.035	0.29 ± 0.018	0.31 ± 0.005	0.31
/m/0335ws	Concert dance	357	0.37 ± 0.096	0.60 ± 0.019	0.60 ± 0.020	0.70
/m/05h35z	Formula racing	351	0.33 ± 0.142	0.55 ± 0.027	0.54 ± 0.048	0.60
/m/01llx7	Bracelet	770	0.09 ± 0.082	0.31 ± 0.042	0.33 ± 0.020	0.24
/m/025sv9y	Landscaping	789	0.26 ± 0.113	0.48 ± 0.120	0.51 ± 0.065	0.63
/m/02qfx	Embroidery	356	0.32 ± 0.044	0.53 ± 0.003	0.53 ± 0.016	0.60
/m/019ctc	Galleon	182	0.45 ± 0.028	0.66 ± 0.005	0.66 ± 0.007	0.61
/m/03lnzv	Calabaza	870	0.50 ± 0.189	0.71 ± 0.025	0.75 ± 0.014	0.81
/m/02zfvv	American shorthair	2084	0.12 ± 0.047	0.32 ± 0.033	0.32 ± 0.038	0.24
/m/01ckgp	Interaction	924	0.04 ± 0.036	0.24 ± 0.058	0.25 ± 0.140	0.37
/m/01lb5	Chess	740	0.53 ± 0.165	0.73 ± 0.007	0.74 ± 0.009	0.86

Table 4: **Middle $\frac{1}{3}$ of classes from Openimages for active learning.** (2 of 3) Average precision for each selected class (153 total) from OpenImages with a labeling budget of 2,000 examples. Classes are in descending order based on the performance gain of MaxEnt-SEALS over Random-All.

Label Name	Display Name	Total Positives	Random (All)	MaxEnt (SEALS)	MaxEnt (All)	Full Supervision
/m/04s5jq	Costume design	818	0.07 ± 0.060	0.26 ± 0.015	0.26 ± 0.019	0.28
/m/02n7rn	Paddy field	468	0.17 ± 0.056	0.36 ± 0.009	0.36 ± 0.011	0.43
/m/05mfbz	Optical instrument	649	0.15 ± 0.071	0.33 ± 0.066	0.33 ± 0.024	0.28
/m/01n5cy	Carpet	644	0.05 ± 0.049	0.23 ± 0.134	0.28 ± 0.122	0.43
/m/02hnl	Drums	741	0.52 ± 0.158	0.70 ± 0.012	0.72 ± 0.010	0.83
/m/0385k	Gymnast	235	0.39 ± 0.070	0.57 ± 0.009	0.59 ± 0.005	0.65
/m/0f70b4	Seafood boil	322	0.31 ± 0.081	0.48 ± 0.002	0.49 ± 0.003	0.51
/m/0hp3n	Pavlova	195	0.19 ± 0.051	0.36 ± 0.008	0.36 ± 0.006	0.34
/m/0fd2m	Deacon	341	0.48 ± 0.057	0.64 ± 0.010	0.64 ± 0.008	0.67
/m/03f0h4	Roman temple	345	0.63 ± 0.037	0.79 ± 0.004	0.79 ± 0.001	0.82
/m/0cffdh	Maple	2301	0.06 ± 0.043	0.22 ± 0.052	0.21 ± 0.014	0.36
/m/07h51	Trail riding	679	0.21 ± 0.031	0.37 ± 0.015	0.37 ± 0.002	0.38
/m/0168g6	Factory	333	0.17 ± 0.070	0.33 ± 0.011	0.34 ± 0.009	0.35
/m/01y2w	Ciconiiformes	426	0.33 ± 0.066	0.49 ± 0.012	0.51 ± 0.012	0.48
/m/0hqkz	Grapefruit	506	0.50 ± 0.062	0.65 ± 0.003	0.65 ± 0.003	0.69
/m/0642b4	Cupboard	898	0.53 ± 0.155	0.68 ± 0.007	0.69 ± 0.007	0.75
/m/03931h	Stele	450	0.12 ± 0.074	0.26 ± 0.021	0.25 ± 0.028	0.35
/m/0cl3b	Rye	128	0.51 ± 0.058	0.64 ± 0.009	0.64 ± 0.007	0.65
/m/0jhv_	Blackberry	245	0.67 ± 0.062	0.80 ± 0.001	0.80 ± 0.001	0.79
/m/02zg0m	Chartreux	147	0.50 ± 0.088	0.63 ± 0.005	0.63 ± 0.005	0.69
/m/03c5wm	Cargo ship	219	0.70 ± 0.134	0.83 ± 0.003	0.83 ± 0.002	0.86
/m/0gv1x	Parrot	1546	0.59 ± 0.085	0.72 ± 0.016	0.76 ± 0.039	0.92
/m/03vd46	Herd	648	0.42 ± 0.200	0.54 ± 0.010	0.55 ± 0.010	0.67
/m/07x7yj	Pancit	385	0.21 ± 0.045	0.33 ± 0.011	0.33 ± 0.007	0.31
/m/025_v	Cactus	377	0.05 ± 0.021	0.17 ± 0.008	0.18 ± 0.011	0.22
/m/01j3sz	Laugh	750	0.06 ± 0.039	0.18 ± 0.074	0.17 ± 0.043	0.26
/m/02zkn_	Log cabin	448	0.44 ± 0.050	0.55 ± 0.006	0.55 ± 0.004	0.62
/m/01fyrh	Downhill	194	0.42 ± 0.065	0.53 ± 0.011	0.51 ± 0.003	0.59
/m/04y4h8h	Bathroom cabinet	368	0.29 ± 0.061	0.40 ± 0.005	0.39 ± 0.007	0.37
/m/079bkr	Mode of transport	1387	0.15 ± 0.232	0.26 ± 0.161	0.16 ± 0.155	0.54
/m/01jnzj	Construction	515	0.13 ± 0.075	0.23 ± 0.012	0.26 ± 0.018	0.34
/m/02qm2x	Icing	1118	0.13 ± 0.054	0.23 ± 0.033	0.25 ± 0.021	0.46
/m/0dlgzb	Bakmi	191	0.27 ± 0.042	0.37 ± 0.021	0.37 ± 0.013	0.36
/m/09l65	Singer	604	0.12 ± 0.050	0.21 ± 0.067	0.21 ± 0.033	0.40
/m/076t48l	Coral reef fish	434	0.51 ± 0.137	0.60 ± 0.007	0.64 ± 0.005	0.79
/m/01295w	Galliformes	674	0.72 ± 0.071	0.80 ± 0.016	0.82 ± 0.026	0.92
/m/0frq6	Pork	464	0.06 ± 0.029	0.14 ± 0.008	0.14 ± 0.006	0.15
/m/0dvg9	Ancient roman architecture	589	0.61 ± 0.109	0.68 ± 0.006	0.70 ± 0.007	0.77
/m/01wkk9	Town square	617	0.31 ± 0.067	0.38 ± 0.016	0.36 ± 0.023	0.47
/m/01_6hg	Delicatessen	196	0.14 ± 0.071	0.21 ± 0.010	0.22 ± 0.010	0.27
/m/03vtj	Ice	682	0.23 ± 0.082	0.30 ± 0.057	0.32 ± 0.009	0.55
/m/02r0zt	White-tailed deer	238	0.34 ± 0.015	0.40 ± 0.028	0.43 ± 0.014	0.43
/m/0hr8	Asphalt	1026	0.23 ± 0.124	0.29 ± 0.026	0.29 ± 0.021	0.45
/m/0b97cn	Lamian	257	0.23 ± 0.139	0.29 ± 0.016	0.32 ± 0.016	0.28
/m/0jqb	Annual plant	677	0.39 ± 0.066	0.44 ± 0.037	0.43 ± 0.043	0.58
/m/019mbd	Antenna	545	0.10 ± 0.040	0.14 ± 0.005	0.13 ± 0.004	0.29
/m/03jqf3	Chevrolet silverado	115	0.05 ± 0.020	0.09 ± 0.012	0.08 ± 0.003	0.12
/m/09q2t	Brown	1427	0.02 ± 0.003	0.05 ± 0.044	0.07 ± 0.069	0.20
/m/06wqb	Space	1006	0.03 ± 0.012	0.06 ± 0.029	0.03 ± 0.044	0.14
/m/01c34b	Flooring	814	0.10 ± 0.010	0.13 ± 0.007	0.14 ± 0.003	0.20
/m/0hlw	Algae	426	0.15 ± 0.063	0.18 ± 0.004	0.19 ± 0.005	0.26

Table 5: **Bottom $\frac{1}{3}$ of classes from Openimages for active learning.** (3 of 3) Average precision for each selected class (153 total) from OpenImages with a labeling budget of 2,000 examples. Classes are in descending order based on the performance gain of MaxEnt-SEALS over Random-All.

Label Name	Display Name	Total Positives	Random (All)	MaxEnt (SEALS)	MaxEnt (All)	Full Supervision
/m/01jfm_	Vehicle registration plate	5697	0.28 ± 0.066	0.31 ± 0.031	0.33 ± 0.048	0.53
/m/09qkx	Boardsport	673	0.26 ± 0.038	0.29 ± 0.011	0.29 ± 0.006	0.53
/m/0322v8	Lugger	103	0.35 ± 0.108	0.37 ± 0.005	0.37 ± 0.016	0.42
/m/03tqj	Icon	186	0.05 ± 0.045	0.07 ± 0.045	0.07 ± 0.064	0.16
/m/05nnm	Organism	1148	0.05 ± 0.041	0.07 ± 0.062	0.13 ± 0.078	0.26
/m/02mk9	Engine	656	0.16 ± 0.046	0.17 ± 0.013	0.17 ± 0.008	0.26
/m/0c0ygc	Stallion	598	0.32 ± 0.126	0.33 ± 0.025	0.40 ± 0.032	0.64
/m/01czv3	Fortification	287	0.43 ± 0.050	0.44 ± 0.025	0.46 ± 0.015	0.52
/m/03hcy1d	Mitsubishi	511	0.01 ± 0.003	0.02 ± 0.003	0.02 ± 0.007	0.04
/m/083jv	White	1494	0.02 ± 0.007	0.03 ± 0.021	0.01 ± 0.003	0.10
/m/07k1x	Tool	1549	0.08 ± 0.016	0.09 ± 0.009	0.10 ± 0.008	0.13
/m/0l14_3	Bowed string instrument	728	0.72 ± 0.111	0.72 ± 0.007	0.74 ± 0.003	0.79
/m/026yps0	East-european shepherd	206	0.61 ± 0.052	0.61 ± 0.001	0.62 ± 0.003	0.65
/m/03jvv0	Teal	975	0.01 ± 0.005	0.01 ± 0.003	0.01 ± 0.001	0.04
/m/0fq22vb	Exhibition	513	0.03 ± 0.032	0.02 ± 0.002	0.02 ± 0.003	0.14
/m/02vwbzz	Electric blue	1180	0.01 ± 0.007	0.00 ± 0.002	0.01 ± 0.002	0.06
/m/01pvk	Canal	726	0.22 ± 0.077	0.21 ± 0.023	0.26 ± 0.006	0.46
/m/0641k	Paper	969	0.16 ± 0.084	0.15 ± 0.056	0.14 ± 0.062	0.41
/m/01w5c_	Aerial photography	931	0.39 ± 0.019	0.37 ± 0.018	0.37 ± 0.015	0.66
/m/0gd2v	Marine mammal	2954	0.19 ± 0.041	0.16 ± 0.014	0.15 ± 0.008	0.21
/m/0br7j6	Pleurotus eryngii	140	0.11 ± 0.024	0.08 ± 0.009	0.08 ± 0.006	0.14
/m/01rh7y	Scale model	667	0.05 ± 0.052	0.02 ± 0.003	0.02 ± 0.003	0.13
/m/01cv4r	Temperate coniferous forest	328	0.30 ± 0.050	0.26 ± 0.016	0.29 ± 0.025	0.40
/m/013y0j	Organ (Biology)	1156	0.23 ± 0.096	0.19 ± 0.095	0.07 ± 0.091	0.44
/m/033kf	Frost	483	0.20 ± 0.123	0.15 ± 0.128	0.21 ± 0.144	0.47
/m/023bbt	Wilderness	1225	0.29 ± 0.105	0.23 ± 0.125	0.24 ± 0.064	0.39
/m/09qqq	Wall	1218	0.11 ± 0.051	0.05 ± 0.015	0.05 ± 0.031	0.27
/m/05qjc	Performing arts	1030	0.12 ± 0.045	0.05 ± 0.020	0.06 ± 0.035	0.53
/m/02pkr5	Plumbing fixture	2124	0.31 ± 0.025	0.24 ± 0.019	0.27 ± 0.019	0.38
/m/038t8_	Estate	667	0.47 ± 0.050	0.39 ± 0.029	0.40 ± 0.006	0.54
/m/04rd7	Mural	649	0.13 ± 0.018	0.05 ± 0.025	0.07 ± 0.029	0.34
/m/0hkvx	Prairie	792	0.37 ± 0.074	0.29 ± 0.032	0.26 ± 0.045	0.57
/m/01d0z1	Shorebird	234	0.32 ± 0.053	0.23 ± 0.008	0.26 ± 0.003	0.37
/m/03nxtz	Cottage	670	0.36 ± 0.214	0.26 ± 0.016	0.36 ± 0.025	0.61
/m/012sbd	Tournament	841	0.15 ± 0.012	0.05 ± 0.031	0.07 ± 0.049	0.16
/m/01dv4h	Portrait	2510	0.23 ± 0.013	0.13 ± 0.076	0.18 ± 0.104	0.43
/m/01f4td	Rural area	921	0.33 ± 0.068	0.22 ± 0.052	0.28 ± 0.042	0.50
/m/0j6jq	Floodplain	567	0.61 ± 0.164	0.47 ± 0.048	0.52 ± 0.037	0.66
/m/047fr	Knitting	409	0.61 ± 0.034	0.46 ± 0.025	0.50 ± 0.006	0.73
/m/0h8ls87	Automotive exterior	1060	0.65 ± 0.122	0.49 ± 0.139	0.54 ± 0.049	0.69
/m/0541p	Multimedia	741	0.45 ± 0.104	0.29 ± 0.029	0.31 ± 0.022	0.53
/m/04qgp	Liqueur	539	0.26 ± 0.020	0.09 ± 0.007	0.14 ± 0.035	0.38
/m/0krfg	Meal	1250	0.52 ± 0.057	0.32 ± 0.074	0.38 ± 0.045	0.59
/m/099md	Soldier	1032	0.62 ± 0.087	0.40 ± 0.029	0.41 ± 0.026	0.72
/m/02mwzg	Plateau	452	0.41 ± 0.047	0.18 ± 0.010	0.24 ± 0.022	0.46
/m/01d7ng	Pelecaniformes	457	0.30 ± 0.122	0.07 ± 0.015	0.09 ± 0.006	0.32
/m/02mnkq	Bumper	985	0.49 ± 0.088	0.25 ± 0.048	0.38 ± 0.047	0.64
/m/03s7w_	Cirque	347	0.43 ± 0.069	0.15 ± 0.113	0.40 ± 0.148	0.55
/m/019cfy	Stadium	1654	0.35 ± 0.040	0.06 ± 0.020	0.10 ± 0.037	0.48
/m/0f6x8	Bird of prey	712	0.76 ± 0.087	0.40 ± 0.090	0.50 ± 0.127	0.91
/m/01sgl	Cycling	794	0.53 ± 0.054	0.13 ± 0.107	0.28 ± 0.142	0.66

A.7 ACTIVE SEARCH ON EACH SELECTED CLASS FROM OPENIMAGES

Table 6: **Top $\frac{1}{3}$ of classes from Openimages for active search.** (1 of 3) Recall (%) of positives for each selected class (153 total) from OpenImages with a labeling budget of 2,000 examples. Classes are in descending order based on the performance gain of MLP-SEALS over Random-All.

Label Name	Display Name	Total Positives	Random (All)	MLP (SEALS)	MLP (All)
/m/01943w	Ibis	259	2.0 ± 0.17	83.9 ± 0.75	83.9 ± 0.80
/m/018gpn	Hurdling	269	1.9 ± 0.00	83.5 ± 0.33	86.2 ± 0.37
/m/02zg0m	Chartreux	147	3.5 ± 0.30	83.9 ± 0.37	84.6 ± 0.78
/m/026yps0	East-european shepherd	206	2.4 ± 0.00	78.2 ± 0.59	78.3 ± 0.43
/m/04y4h8h	Bathroom cabinet	368	1.4 ± 0.12	76.8 ± 0.35	77.1 ± 0.23
/m/0jhv_	Blackberry	245	2.0 ± 0.00	77.5 ± 0.45	78.5 ± 0.74
/m/06lk1	Rat	1151	0.5 ± 0.08	75.1 ± 0.16	75.2 ± 0.26
/m/026545k	Elk	353	1.5 ± 0.16	73.4 ± 0.31	74.3 ± 0.81
/m/0cl3b	Rye	128	3.9 ± 0.00	74.7 ± 1.18	74.5 ± 1.52
/m/0f70b4	Seafood boil	322	1.6 ± 0.00	70.4 ± 0.42	70.6 ± 0.52
/m/0hp3n	Pavlova	195	2.6 ± 0.00	70.8 ± 0.36	71.3 ± 0.96
/m/03f0h4	Roman temple	345	1.5 ± 0.13	69.2 ± 0.53	68.3 ± 0.70
/m/03d8m_	Monster truck	286	1.7 ± 0.00	68.1 ± 0.76	67.8 ± 0.52
/m/01fyrh	Downhill	194	2.6 ± 0.00	67.2 ± 0.59	69.0 ± 0.23
/m/01d0z1	Shorebird	234	2.1 ± 0.00	66.8 ± 0.56	66.4 ± 0.38
/m/02yhp1	San Pedro cactus	318	1.6 ± 0.14	65.8 ± 0.41	64.9 ± 0.28
/m/09wzfj	California roll	368	1.4 ± 0.12	65.3 ± 0.49	68.0 ± 0.12
/m/0cghn	Mcdonnell douglas f/a-18 hornet	160	3.2 ± 0.28	66.0 ± 5.67	67.9 ± 5.35
/m/0br7j6	Pleurotus eryngii	140	3.6 ± 0.00	65.7 ± 1.43	66.1 ± 1.08
/m/0385k	Gymnast	235	2.2 ± 0.19	64.0 ± 0.65	64.0 ± 0.71
/m/019ctc	Galleon	182	2.7 ± 0.00	62.4 ± 0.92	61.5 ± 1.35
/m/07h51	Trail riding	679	0.8 ± 0.08	59.7 ± 0.51	60.7 ± 0.47
/m/03c5wm	Cargo ship	219	2.3 ± 0.00	61.1 ± 0.75	61.7 ± 0.60
/m/0hqkz	Grapefruit	506	1.0 ± 0.09	59.4 ± 0.27	60.4 ± 0.33
/m/01ngsj	Daylily	391	1.3 ± 0.11	59.4 ± 0.80	59.5 ± 1.01
/m/01cz4w	Bilberry	228	2.2 ± 0.00	58.9 ± 1.93	55.2 ± 2.31
/m/058hry	Smoothie	330	1.5 ± 0.00	58.0 ± 1.30	59.8 ± 0.78
/m/02qfx	Embroidery	356	1.5 ± 0.13	57.6 ± 0.61	57.2 ± 1.04
/m/0fd2m	Deacon	341	1.5 ± 0.13	57.1 ± 0.79	57.9 ± 0.44
/m/05h5q_s	Glider	393	1.3 ± 0.00	55.8 ± 0.82	57.6 ± 0.95
/m/04l3lm	Shooting range	189	2.6 ± 0.00	56.3 ± 1.43	55.6 ± 3.45
/m/02r0zt	White-tailed deer	238	2.2 ± 0.19	55.8 ± 1.50	55.9 ± 1.73
/m/076t48l	Coral reef fish	434	1.3 ± 0.21	54.8 ± 0.75	54.9 ± 0.41
/m/07x7yj	Pancit	385	1.3 ± 0.00	52.8 ± 0.56	53.1 ± 0.30
/m/01lb5	Chess	740	0.7 ± 0.00	51.9 ± 0.70	50.9 ± 1.23
/m/03jqf3	Chevrolet silverado	115	4.3 ± 0.00	54.1 ± 2.64	54.6 ± 2.78
/m/058qzx	Kitchen knife	360	1.5 ± 0.15	50.9 ± 0.64	53.9 ± 0.44
/m/0cl6l	Straw	547	1.0 ± 0.10	50.3 ± 0.33	51.0 ± 0.55
/m/0dlgzb	Bakmi	191	2.6 ± 0.00	51.8 ± 0.83	51.2 ± 0.68
/m/0322v8	Lugger	103	4.9 ± 0.00	53.8 ± 1.62	53.8 ± 1.47
/m/0dvg9	Ancient roman architecture	589	0.8 ± 0.00	48.5 ± 0.96	47.1 ± 1.03
/m/019mbd	Antenna	545	1.0 ± 0.08	47.4 ± 0.21	48.0 ± 0.65
/m/0b97cn	Lamian	257	1.9 ± 0.00	47.8 ± 1.11	48.2 ± 0.64
/m/03lnzv	Calabaza	870	0.6 ± 0.05	46.0 ± 0.35	45.8 ± 0.34
/m/09l0j_	Ring	676	0.7 ± 0.00	45.2 ± 0.47	45.4 ± 0.39
/m/01y2w	Ciconiiformes	426	1.2 ± 0.10	45.2 ± 0.82	45.2 ± 0.49
/m/02zkn_	Log cabin	448	1.1 ± 0.00	44.9 ± 1.20	45.7 ± 0.46
/m/0l14_3	Bowed string instrument	728	0.7 ± 0.06	44.4 ± 0.32	44.7 ± 0.42
/m/05z55	Pasta	954	0.5 ± 0.00	43.7 ± 0.34	43.8 ± 0.51
/m/047fr	Knitting	409	1.3 ± 0.11	43.5 ± 0.72	42.8 ± 1.09
/m/01xc8d	Rope	618	0.8 ± 0.00	43.0 ± 0.39	42.8 ± 0.25

Table 7: **Middle $\frac{1}{3}$ of classes from Openimages for active search.** (2 of 3) Recall (%) of positives for each selected class (153 total) from OpenImages with a labeling budget of 2,000 examples. Classes are in descending order based on the performance gain of MLP-SEALS over Random-All.

Label Name	Display Name	Total Positives	Random (All)	MLP (SEALS)	MLP (All)
/m/02n7rn	Paddy field	468	1.1 ± 0.10	42.6 ± 0.57	44.2 ± 1.05
/m/05h35z	Formula racing	351	1.4 ± 0.00	42.6 ± 0.74	41.4 ± 0.37
/m/02mk9	Engine	656	0.8 ± 0.00	41.7 ± 0.46	40.6 ± 1.81
/m/0ll1f78	Shrimp	907	0.6 ± 0.00	40.4 ± 0.55	40.8 ± 0.48
/m/01s0ps	Electric piano	345	1.5 ± 0.13	40.9 ± 1.73	42.1 ± 0.78
/m/03fwl	Goat	1190	0.4 ± 0.00	39.6 ± 0.57	39.6 ± 0.23
/m/0642b4	Cupboard	898	0.6 ± 0.06	39.6 ± 0.46	39.6 ± 0.34
/m/0gv1x	Parrot	1546	0.4 ± 0.08	39.2 ± 0.30	38.8 ± 0.28
/m/0dqb5	Citrus	796	0.7 ± 0.11	39.3 ± 0.61	39.6 ± 0.09
/m/0cfk35	Chocolate truffle	288	1.8 ± 0.16	39.6 ± 2.56	39.9 ± 2.19
/m/0dxb5	Berry	874	0.6 ± 0.05	37.8 ± 0.47	37.6 ± 0.30
/m/01ll1r	Briefs	539	1.0 ± 0.08	37.1 ± 0.36	37.2 ± 0.57
/m/01_6hg	Delicatessen	196	2.6 ± 0.00	38.2 ± 0.76	39.0 ± 0.46
/m/0dwxr	Modern pentathlon	772	0.6 ± 0.00	35.9 ± 5.45	32.6 ± 12.85
/m/0335ws	Concert dance	357	1.4 ± 0.00	36.6 ± 0.67	36.1 ± 0.69
/m/0c0ygc	Stallion	598	0.9 ± 0.09	35.7 ± 0.56	36.3 ± 0.70
/m/0176mf	Belt	467	1.1 ± 0.00	35.2 ± 0.35	34.9 ± 0.48
/m/01czv3	Fortification	287	1.7 ± 0.00	35.7 ± 1.75	37.6 ± 0.90
/m/01295w	Galliformes	674	0.7 ± 0.00	33.9 ± 0.67	33.9 ± 0.62
/m/03931h	Stele	450	1.1 ± 0.00	33.9 ± 0.96	32.7 ± 1.46
/m/03vd46	Herd	648	0.8 ± 0.07	33.5 ± 0.44	33.7 ± 0.35
/m/0hlw	Algae	426	1.2 ± 0.00	33.8 ± 1.30	33.1 ± 1.24
/m/0h8n982	Shelving	810	0.7 ± 0.15	33.2 ± 0.31	33.3 ± 0.11
/m/01d7ng	Pelecaniformes	457	1.1 ± 0.10	33.4 ± 9.74	37.5 ± 0.40
/m/02hnl	Drums	741	0.7 ± 0.00	32.9 ± 0.40	32.7 ± 0.25
/m/04fyb1	Sirloin steak	297	1.8 ± 0.15	33.9 ± 0.99	32.7 ± 1.13
/m/025_v	Cactus	377	1.3 ± 0.00	33.4 ± 1.57	35.2 ± 1.33
/m/020dp	Cranberry	450	1.2 ± 0.10	32.9 ± 0.83	33.7 ± 0.81
/m/0168g6	Factory	333	1.5 ± 0.00	32.0 ± 0.81	31.7 ± 1.25
/m/04s5jq	Costume design	818	0.6 ± 0.00	30.9 ± 1.45	30.6 ± 1.17
/m/05mfbz	Optical instrument	649	0.8 ± 0.00	30.3 ± 5.72	32.8 ± 0.52
/m/01jnzj	Construction	515	1.0 ± 0.00	30.1 ± 0.90	31.1 ± 0.57
/m/01cv4r	Temperate coniferous forest	328	1.5 ± 0.00	30.1 ± 0.77	27.6 ± 2.75
/m/033cnk	Egg (Food)	1193	0.4 ± 0.00	28.8 ± 3.17	28.6 ± 3.04
/m/0wgq98_	Skating	561	1.0 ± 0.16	28.8 ± 5.85	30.4 ± 1.88
/m/02pkr5	Plumbing fixture	2124	0.3 ± 0.05	27.9 ± 0.48	27.9 ± 0.28
/m/0g9vs81	Steamed rice	580	0.9 ± 0.00	28.1 ± 1.48	30.2 ± 1.37
/m/0cyz6w	Whole food	708	0.7 ± 0.00	27.7 ± 0.86	27.5 ± 0.44
/m/09qkx	Boardsport	673	0.8 ± 0.07	26.8 ± 0.20	26.5 ± 0.66
/m/0frq6	Pork	464	1.1 ± 0.00	26.3 ± 1.03	26.6 ± 1.65
/m/01w5c_	Aerial photography	931	0.6 ± 0.06	25.8 ± 0.26	26.1 ± 0.40
/m/01wkk9	Town square	617	0.8 ± 0.00	25.7 ± 0.90	26.1 ± 0.82
/m/0cffdh	Maple	2301	0.2 ± 0.02	24.3 ± 0.53	24.4 ± 0.50
/m/038t8_	Estate	667	0.9 ± 0.13	24.8 ± 1.22	25.9 ± 0.54
/m/01xq0k1	Cattle	5995	0.1 ± 0.01	23.8 ± 0.13	23.6 ± 0.32
/m/0btmb	Superhero	968	0.6 ± 0.06	23.4 ± 3.08	23.3 ± 1.42
/m/01llx7	Bracelet	770	0.6 ± 0.00	23.2 ± 5.38	24.8 ± 0.29
/m/033kf	Frost	483	1.0 ± 0.00	23.1 ± 1.93	22.5 ± 2.67
/m/01rh7y	Scale model	667	0.8 ± 0.13	22.9 ± 1.02	23.7 ± 4.30
/m/0f6x8	Bird of prey	712	0.7 ± 0.00	22.4 ± 0.66	22.0 ± 0.51
/m/01pvk	Canal	726	0.7 ± 0.06	22.4 ± 0.46	20.9 ± 1.09

Table 8: **Bottom $\frac{1}{3}$ of classes from Openimages for active search.** (3 of 3) Recall (%) of positives for each selected class (153 total) from OpenImages with a labeling budget of 2,000 examples. Classes are in descending order based on the performance gain of MLP-SEALS over Random-All.

Label Name	Display Name	Total Positives	Random (All)	MLP (SEALS)	MLP (All)
/m/02mwzg	Plateau	452	1.1 ± 0.00	22.7 ± 0.50	19.1 ± 3.54
/m/01n5cy	Carpet	644	0.8 ± 0.07	21.9 ± 5.41	22.7 ± 0.95
/m/0j_5b	Monoplane	756	0.7 ± 0.07	21.8 ± 2.08	20.1 ± 1.43
/m/0fq22vb	Exhibition	513	1.0 ± 0.09	21.9 ± 0.45	23.1 ± 0.38
/m/03vtj	Ice	682	0.8 ± 0.07	21.6 ± 1.04	23.1 ± 3.43
/m/0cnmr	Fur	834	0.6 ± 0.07	21.2 ± 0.97	17.3 ± 9.06
/m/02qm2x	Icing	1118	0.4 ± 0.00	20.5 ± 0.59	20.1 ± 0.26
/m/01c34b	Flooring	814	0.6 ± 0.05	20.4 ± 0.46	16.9 ± 8.97
/m/0hkvx	Prairie	792	0.6 ± 0.00	19.0 ± 1.00	19.2 ± 0.50
/m/0cnxs6x	Tooth	976	0.5 ± 0.05	18.6 ± 2.01	18.0 ± 2.36
/m/0bpn3c2	Skateboarding Equipment	862	0.6 ± 0.05	18.1 ± 6.15	19.3 ± 4.83
/m/0h8ls87	Automotive exterior	1060	0.5 ± 0.04	17.7 ± 2.93	11.9 ± 10.03
/m/03tqj	Icon	186	2.7 ± 0.00	19.9 ± 0.85	17.2 ± 4.82
/m/0gd2v	Marine mammal	2954	0.2 ± 0.03	17.3 ± 0.45	17.2 ± 0.46
/m/03nxtz	Cottage	670	0.7 ± 0.00	17.6 ± 0.57	17.3 ± 0.47
/m/099md	Soldier	1032	0.5 ± 0.09	17.3 ± 0.69	16.8 ± 0.64
/m/07k1x	Tool	1549	0.3 ± 0.03	17.0 ± 0.64	16.9 ± 0.65
/m/02zfvv	American shorthair	2084	0.3 ± 0.03	16.5 ± 0.42	16.7 ± 0.59
/m/0541p	Multimedia	741	0.7 ± 0.06	16.8 ± 2.04	17.1 ± 0.99
/m/0hr8	Asphalt	1026	0.5 ± 0.04	15.1 ± 0.71	11.5 ± 5.95
/m/09l65	Singer	604	0.9 ± 0.07	14.6 ± 7.10	13.6 ± 7.03
/m/0j6jq	Floodplain	567	0.9 ± 0.00	14.6 ± 1.33	14.0 ± 1.40
/m/01f4td	Rural area	921	0.6 ± 0.05	14.2 ± 0.36	13.2 ± 0.91
/m/03hcy1d	Mitsubishi	511	1.0 ± 0.09	12.6 ± 0.30	11.8 ± 1.11
/m/013y0j	Organ (Biology)	1156	0.5 ± 0.12	12.1 ± 8.09	15.9 ± 5.12
/m/0641k	Paper	969	0.5 ± 0.00	12.0 ± 6.63	14.8 ± 1.73
/m/0jqb	Annual plant	677	0.7 ± 0.00	11.8 ± 1.16	10.7 ± 2.90
/m/02vwbzz	Electric blue	1180	0.5 ± 0.05	11.5 ± 1.07	9.4 ± 4.96
/m/019cfy	Stadium	1654	0.3 ± 0.03	10.8 ± 0.83	9.3 ± 2.35
/m/04rd7	Mural	649	0.8 ± 0.14	10.4 ± 2.13	10.3 ± 1.19
/m/03jvv0	Teal	975	0.5 ± 0.05	9.9 ± 0.46	10.4 ± 0.27
/m/09qqq	Wall	1218	0.4 ± 0.04	9.3 ± 8.06	12.0 ± 6.49
/m/01bmhj	Thumb	895	0.6 ± 0.06	9.3 ± 7.45	13.8 ± 2.69
/m/01jfm_	Vehicle registration plate	5697	0.1 ± 0.01	8.7 ± 0.99	8.3 ± 2.07
/m/025sv9y	Landscaping	789	0.7 ± 0.06	9.2 ± 2.03	9.3 ± 1.41
/m/03s7w_	Cirque	347	1.5 ± 0.13	9.9 ± 2.53	9.8 ± 1.88
/m/0krfg	Meal	1250	0.4 ± 0.07	8.5 ± 2.97	9.1 ± 2.57
/m/023bbt	Wilderness	1225	0.4 ± 0.00	8.5 ± 4.58	9.8 ± 2.46
/m/06wqb	Space	1006	0.5 ± 0.04	7.8 ± 5.14	6.3 ± 5.74
/m/04qgp	Liqueur	539	1.0 ± 0.17	8.0 ± 6.36	12.8 ± 4.37
/m/09q2t	Brown	1427	0.4 ± 0.03	7.2 ± 5.31	2.6 ± 4.79
/m/01sgl	Cycling	794	0.6 ± 0.00	7.3 ± 2.08	7.8 ± 2.42
/m/05nnm	Organism	1148	0.4 ± 0.00	6.8 ± 3.70	2.0 ± 2.82
/m/01j3sz	Laugh	750	0.7 ± 0.06	6.6 ± 2.98	8.6 ± 1.07
/m/01dv4h	Portrait	2510	0.2 ± 0.02	5.8 ± 0.98	5.3 ± 0.70
/m/02mnkq	Bumper	985	0.5 ± 0.06	5.9 ± 4.01	8.3 ± 3.49
/m/079bkr	Mode of transport	1387	0.4 ± 0.04	5.1 ± 2.18	3.6 ± 2.87
/m/01ckgp	Interaction	924	0.6 ± 0.12	4.5 ± 3.72	4.6 ± 2.64
/m/012sbd	Tournament	841	0.6 ± 0.07	4.3 ± 2.23	5.1 ± 1.23
/m/05qjc	Performing arts	1030	0.5 ± 0.00	2.3 ± 0.89	2.5 ± 0.77
/m/083jv	White	1494	0.3 ± 0.00	2.0 ± 1.19	0.5 ± 0.16

A.8 SELF-SUPERVISED EMBEDDING (SENTENCE-BERT) ON GOODREADS

We followed the same general procedure described in Section 4.1, aside from the dataset specific details below. Goodreads spoiler detection (Wan et al., 2019) had 17.67 million sentences with binary spoiler annotations. Spoilers made up 3.224% of the data, making them much more common than the rare concepts we evaluated in the other datasets. Following Wan et al. (2019), we used 3.53 million sentences for testing (20%), 10,000 sentences as the validation set, and the remaining 14.13 million sentences as the unlabeled pool. We also switched to the area under the ROC curve (AUC) as our primary evaluation metric for active learning to be consistent with Wan et al. (2019). For G_z, we used a pre-trained Sentence-BERT model (SBERT-NLI-base) (Reimers & Gurevych, 2019), applied PCA whitening to reduce the dimension to 256, and performed l^2 normalization.

A.8.1 ACTIVE SEARCH

SEALS achieved the same recall as the baseline approaches, but only considered less than 1% of the unlabeled data in the candidate pool, as shown in Figure 15. At a labeling budget of 2,000, MLP-ALL and MLP-SEALS recalled $0.15 \pm 0.02\%$ and $0.17 \pm 0.05\%$, respectively, while MaxEnt-All and MaxEnt-SEALS achieved $0.14 \pm 0.04\%$ and $0.11 \pm 0.06\%$ recall respectively. Increasing the labeling budget to 50,000 examples, increased recall to ˜3.7% for MaxEnt and MLP but maintained a similar relative improvement over random sampling, as shown in Figure 16. ID-SEALS performed worse than the other strategies. However, all of the active selection strategies outperformed random sampling by up to an order of magnitude.

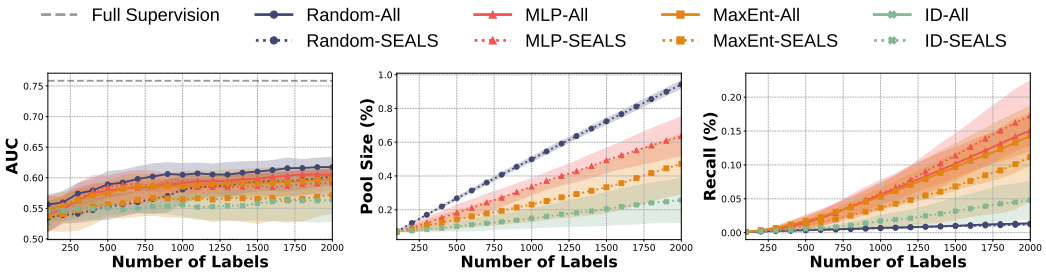


Figure 15: Active learning and search on Goodreads with Sentence-BERT embeddings. Across datasets and strategies, SEALS with $k = 100$ performs similarly to the baseline approach in terms of both the error the model achieves for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the data U (middle).

A.8.2 ACTIVE LEARNING

At a labeling budget of 2,000 examples, all the selection strategies were indistinguishable from random sampling. Increasing the labeling budget did not help, as shown in Figure 16. Unlike ImageNet and OpenImages, Goodreads had a much higher fraction of positive examples (3.224%), and the examples were not tightly clustered as described in Section A.8.3.

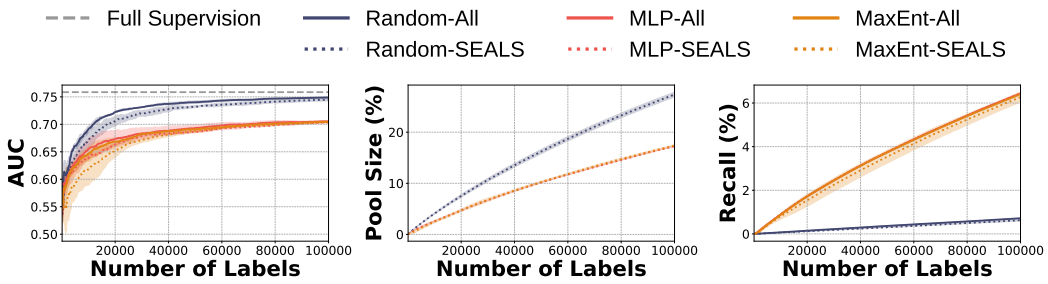


Figure 16: Active learning and search on Goodreads with a labeling budget of 100,000 examples. Across strategies, SEALS with $k = 100$ performed similarly to the baseline approach in terms of both the error the model achieved for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the data U (middle). ID was excluded because of the growing pool size and computation. For active search, MaxEnt and MLP continued to improve recall. For active learning, all the selection strategies (both with and without SEALS) performed worse than random sampling despite the larger labeling budget. This gap was likely due to spoilers being book specific and the higher fraction of positive examples in the unlabeled pool, causing relevant examples to be spread almost uniformly across the space (see Section A.8.3).

A.8.3 Latent structure

The large number of positive examples in the Goodreads dataset limited the analysis we could perform. We could only calculate the size of the largest connected component in the nearest neighbor graph (Figure 17). For $k = 10$, only 28.4% of the positive examples could be reached directly, but increasing k to 100 improved that dramatically to 96.7%. For such a large connected component, one might have expected active learning to perform better in Section A.8.2. By analyzing the embeddings, however, we found that examples are spread almost uniformly across the space with an average cosine similarity of 0.004. For comparison, the average cosine similarity for concepts in ImageNet and OpenImages was 0.453 ± 0.077 and 0.361 ± 0.105 respectively. This uniformity was likely due to the higher fraction of positive examples and spoilers being book specific while Sentence-BERT is trained on generic data. As a result, even if spoilers were tightly clustered within each book, the books were spread across a range of topics and consequently across the embedding space, illustrating a limitation and opportunity for future work.

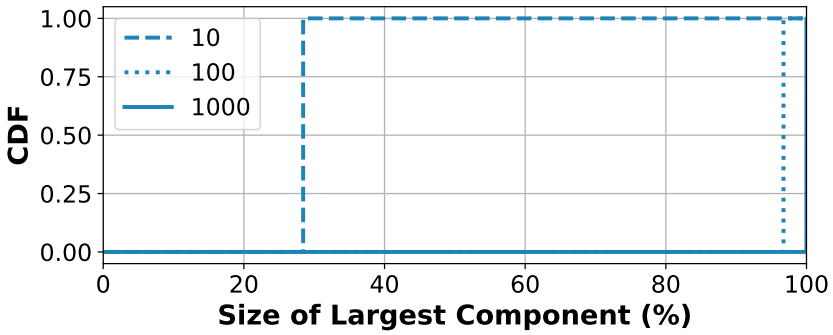


Figure 17: Cumulative distribution function (CDF) for the largest connected component in the Goodreads dataset with varying values of k.