Similarity Search for Efficient Active Learning and Search of Rare Concepts

Cody Coleman, [1] Edward Chou, [2] Julian Katz-Samuels, [3] Sean Culatana, [2] Peter Bailis, [1] Alexander C. Berg, [4] Robert Nowak, [3] Roshan Sumbaly, [2] Matei Zaharia, [1] I. Zeki Yalniz [2]

[1]Stanford University, [2]Facebook AI, [3]University of Wisconsin, [4]Facebook AI Research
cody@cs.stanford.edu

Abstract

Many active learning and search approaches are intractable for large-scale industrial settings with billions of unlabeled examples. Existing approaches search globally for the optimal examples to label, scaling linearly or even quadratically with the unlabeled data. In this paper, we improve the computational efficiency of active learning and search methods by restricting the candidate pool for labeling to the nearest neighbors of the currently labeled set instead of scanning over all of the unlabeled data. We evaluate several selection strategies in this setting on three large-scale computer vision datasets: ImageNet, OpenImages, and a de-identified and aggregated dataset of 10 billion publicly shared images provided by a large internet company. Our approach achieved similar mAP and recall as the traditional global approach while reducing the computational cost of selection by up to three orders of magnitude, enabling *web-scale active learning*.

1 Introduction

Large-scale unlabeled datasets can contain millions or billions of examples covering a wide variety of underlying concepts [Chelba et al. 2013, Zhang, Zhao, and LeCun 2015, Wan et al. 2019, Russakovsky et al. 2015, Kuznetsova et al. 2020, Thomee et al. 2016, Abu-El-Haija et al. 2016, Lee, Rao, and Arnold 2019]. Yet, these massive datasets often skew towards a relatively small number of common concepts, for example 'cats', 'dogs', and 'people' [Liu et al. 2019, Zhang et al. 2017, Wang, Ramanan, and Hebert 2017, Van Horn and Perona 2017]. Rare concepts, such as 'harbor seals', tend to only appear in a small fraction of the data (usually less than 1%). However, performance on these rare concepts is critical in many settings. For example, harmful or malicious content may comprise only a small percentage of user-generated content, but it can have a disproportionate 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, Yi, and Chao 2019, Karpathy 2020]. Even a simple task, such as stop sign detection by an autonomous vehicle, can be difficult due to 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), can be heavily occluded, or have modifiers (e.g., "Except Right Turn") [Karpathy 2020]. Large-scale datasets are essential but not sufficient; finding the relevant examples for these long-tail tasks is challenging.

Active learning and search methods have the potential to automate the process of identifying these rare, high-value data points, but often become intractable at-scale. Existing techniques carefully select examples over a series of rounds to improve model quality (active learning [Settles 2012]) or find positive examples in highly skewed settings (active search [Garnett et al. 2012]). Each selection round iterates over the entire unlabeled data to identify the optimal example or batch of examples to label based on uncertainty (e.g., the entropy of predicted class probabilities) or other heuristics [Settles 2011, 2012, Lewis and Gale 1994, Garnett et al. 2012, Zhang et al. 2020, Beluch et al. 2018, Yoo and Kweon 2019, Gal, Islam, and Ghahramani 2017, He and Carbonell 2007, Sinha, Ebrahimi, and Darrell 2019, Joshi, Porikli, and Papanikolopoulos 2009, Settles and Craven 2008, Sener and Savarese 2018]. Depending on the selection criteria, each round can scale linearly [Lewis and Gale 1994, Joshi, Porikli, and Papanikolopoulos 2009] or even quadratically [Settles and Craven 2008, Sener and Savarese 2018] with the size of the unlabeled data. The computational cost of this process has become an impediment as datasets and model architectures have increased rapidly in size [Amodei and 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 and Savarese 2018, Kirsch, van Amersfoort, and Gal 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 batch for each round, which remains intractable for web-scale datasets with billions of examples. The selection rounds of these techniques need to scale sublinearly with the unlabeled data size to tackle these massive and heavily skewed problems.

In this paper, we propose Similarity search for Efficient Active Learning and Search (SEALS) as a simple approach to further improve computational efficiency and achieve *web-scale active learning*. Empirically, we find that learned representations from pre-trained models can effectively cluster many unseen 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 rather than scanning over all of the unlabeled data. Finding the nearest neighbors for each labeled example in the unlabeled data can be performed efficiently with sublinear retrieval times [Charikar 2002] and sub-second latency on billion-scale datasets [Johnson, Douze, and Jégou 2017] for approximate approaches. While this restricted candidate pool of unlabeled examples impacts theoretical sample complexity, our analysis shows that SEALS still achieves the optimal logarithmic dependence on the desired error for active learning. As a result, SEALS maintains similar label-efficiency and enables selection to scale with the size of the labeled data and only sublinearly with the size of the unlabeled data, making active learning and search tractable on web-scale datasets with billions of 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 de-identified and aggregated dataset of 10 billion publicly shared 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 and Gale 1994], information density [Settles and 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 reducing the computational cost by up to three orders of magnitude. Consequently, SEALS could perform several selection rounds over 10 billion images in seconds with a single machine, unlike the baselines that needed a cluster with tens of thousands of cores. To our knowledge, no other works have performed active learning at this scale.

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. Much of the 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 and Savarese 2018, Kirsch, van Amersfoort, and Gal 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, but still require at least linear work over the whole unlabeled set for each selection round. Our work reduces the number of examples considered in each selection round such that active learning scales sublinearly with the size of the unlabeled dataset.

Others have tried to improve computational efficiency by using much smaller models as cheap proxies, generating examples, or subsampling data. A smaller model reduces the computation required per example [Yoo and Kweon 2019, Coleman et al. 2020]; but unlike our approach, it still requires passing over all of the unlabeled examples. Generative approaches [Mayer and Timofte 2020, Zhu and Bento 2017, Lin, Mausam, and Weld 2018] enable sublinear selection runtime complexities; but they struggle to match the label-efficiency of traditional approaches due to the highly variable quality of the generated examples. Subsampling the unlabeled data as in [Ertekin et al. 2007] also avoids iterating over all of the data. However, for rare concepts in web-scale datasets, randomly chosen examples are extremely unlikely to be close enough to the decision boundary. Our work both achieves sublinear selection runtimes and matches the label-efficiency of traditional approaches.

There are also specific optimizations for certain families of models. Jain, Vijayanarasimhan, and Grauman [2010] developed custom hashing schemes for the weights from linear SVM classifiers to efficiently find examples near the decision boundary. While this enables a sublinear selection runtime complexity similar to SEALS, the hashing hyperplanes approach is non-trivial to generalize to other families of models and even SVMs with non-linear kernels. Our work extends to a wide variety of models and selection strategies.

k-nearest neighbor (k-NN) classifiers are also advantageous because they do not require an explicit training phase [He and Carbonell 2007, Joshi, Porikli, and Papanikolopoulos 2012, Wei, Iyer, and Bilmes 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. These approaches still require evaluating all of the data, which can be prohibitively expensive on large-scale datasets. Our work targets the selection phase rather than training and uses k-NNs to limit candidate examples and not as a classifier.

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, Jiang, Garnett, and Moseley 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 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 vision and text where the available unlabeled datasets are much larger, and computational efficiency is a significant bottleneck.

3 Problem Statement

This section formally outlines the problems of active learning (Section 3.1) and search (Section 3.2) as well as the selection methods we evaluated. For both, we examined the pool-based batch setting, where examples are selected in batches to improve computational efficiency.

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 one 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 and Gale 1994]:

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

and **information density (ID)** [Settles and Craven 2008]:

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

where $\text{sim}(\mathbf{z}, \mathbf{z}_p)$ is the cosine similarity of the embedded examples and $\beta = 1$. Note that for binary classification, MaxEnt is equivalent to least confidence and margin sampling, which are also popular criteria for uncertainty sampling [Settles 2009]. While MaxEnt uncertainty sampling only requires a linear pass over the unlabeled data, ID scales quadratically with $|U|$ because it weighs each example's informativeness by its similarity to all other examples. To improve computational performance, the average similarity scores can be cached after the first round so that subsequent rounds scale linearly.

We explored the greedy k-centers approach from Sener and 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.

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]:

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

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

Algorithm 1: BASELINE APPROACH

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

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

Algorithm 2: SEALS APPROACH

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

4 Similarity search for Efficient Active Learning and Search (SEALS)

This section describes SEALS and how it improves computational efficiency and impacts sample complexity. As shown in Algorithm 2, SEALS makes two modifications to accelerate the inner loop of Algorithm 1:

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 $\mathcal{P}_r$.

Both modifications can be done transparently for many selection strategies, making SEALS applicable to a wide range of methods, even beyond the ones considered here.

By restricting the candidate pool to the labeled examples' nearest neighbors, SEALS applies the selection strategy to at most $k|L_r|$ examples. Finding the k nearest neighbors for each labeled example adds overhead, but it can be calculated efficiently with sublinear retrieval times [Charikar 2002] and sub-second latency on billion-scale datasets [Johnson, Douze, and Jégou 2017] for approximate approaches.

Computational savings. Each selection round scales with the size of the labeled dataset and sublinearly with the size of the unlabeled data. 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 bound 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. However, this cost amortizes over many selection rounds, concepts, or other applications. Similarity search is a critical workload for information retrieval and powers many applications, including recommendation, with deep learning embeddings increasingly being used [Babenko et al. 2014, Babenko and Lempitsky 2016, Johnson, Douze, and Jégou 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 to avoid additional costs.

Sample complexity. To shed light on why SEALS works, we analyzed an idealized setting where classes are linearly separable and examples are already embedded ($\mathbf{x} = G_z(\mathbf{x})$). Let $\mathcal{X} \subset \mathbb{R}^d$ be some convex set and $\mathbf{w}_* \in \mathbb{R}^d$. An example $\mathbf{x} \in \mathcal{X}$ has a label $y = 1$ if $\mathbf{x}^\top \mathbf{w}_* \geq 0$ and a label $y = -1$ otherwise. We assume that the k nearest neighbor graph $\mathcal{G} = (\mathcal{X}, E)$ satisfies the property that for each $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, if $\|\mathbf{x} - \mathbf{x}'\|_2 \leq \delta$, then $(\mathbf{x}, \mathbf{x}') \in E$, so any point in a ball around an example $\mathbf{x}$ is a neighbor of $\mathbf{x}$. We also assume that the algorithm is given n_0 labeled seeds points $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_0}\} \subset \mathcal{X}$ where $n_0 \geq d - 1$. To prove a result, we consider a slightly modified version of SEALS that performs $d - 1$ parallel nearest neighbor searches, each one initiated with one of the seed points $\mathbf{x}_i$ with $i \in \{1, \ldots, d-1\}$, (see the supplementary material for a formal description and a proof). Note, this procedure still aligns with the batch queries in SEALS.

Theorem 1. *Let $\epsilon > 0$ and let γ_i denote the distance from the seed $\mathbf{x}_i$ to the convex hull of oppositely labeled seed points. There exists a constant $\sigma > 0$ that quantifies the diversity of the seeds (defined below) such that after SEALS makes $O(\max_{i \in \{1,\ldots,d-1\}} d(\frac{\gamma_i}{\delta} + \log(\frac{d\delta}{\epsilon \min(\sigma,1)})))$ queries, its estimate $\widehat{\mathbf{w}} \in \mathbb{R}^d$ satisfies $\|\widehat{\mathbf{w}} - \mathbf{w}_*\|_2 \leq \epsilon$.*

The sample complexity bound compares favorably to known optimal sample complexities in this setting [Balcan and Long 2013]: $O(d/\epsilon)$ and $O(d \log(1/\epsilon))$ for passive and active learning, respectively. In particular, the SEALS bound has the optimal logarithmic dependence on ϵ.

The parameter γ_i is an upper bound on the distance of $\mathbf{x}_i$ to the true decision boundary. Let B_i denote the ball of radius $\gamma_i + 2\delta + \epsilon$ centered at $\mathbf{x}_i$, where $\epsilon > 0$ is fixed. The true decision boundary must intersect B_i. Let $\mathcal{Z}_i \subset B_i$ denote the set of points in B_i that are within ϵ of the boundary. The constant σ is a measure of the diversity of the seed examples, defined as:

$$\sigma = \min_{\mathbf{z}_i \in \mathcal{Z}_i : i \in \{1,\ldots,d-1\}} \sigma_{d-1}([\mathbf{z}_1 \ \cdots, \ \mathbf{z}_{d-1}])$$

where $\sigma_{d-1}(\cdot)$ is the $(d-1)$th singular value of the matrix. If the $\mathcal{Z}_i$ sets are well separated and if the centers form a well-conditioned basis for a $(d-1)$-dimensional subspace in $\mathbb{R}^d$, then σ is a reasonable constant.

Intuitively, the algorithm has two phases: a slow phase and a fast phase. During the slow phase, the algorithm queries points that slowly approach the true decision boundary at a rate δ. After at most $O(\max_i d\frac{\gamma_i}{\delta})$ queries, the algorithm finds $d-1$ points that are within δ of the true decision boundary and enters the fast phase. Since the algorithm has already found points that are close to the decision boundary, the constraints of the nearest neighbor graph essentially do not encumber the algorithm, enabling it to home in on the true decision boundary at an exponential rate of $O(d \log(\frac{d\delta}{\epsilon\sigma})))$.

5 Experiments

We applied SEALS to three selection strategies (MaxEnt, MLP, and ID) and performed active learning and search on three separate datasets: ImageNet [Russakovsky et al. 2015], OpenImages [Kuznetsova et al. 2020], and a de-identified and aggregated dataset of 10 billion publicly shared images (Table 1). Section 5.1 details the experimental setup used for both the baselines that run over all of the data (*-All) and our proposed method that restricts the candidate pool (*-SEALS). Sections 5.2, 5.3, and 5.4 provide dataset-specific details and present the active learning and search results for ImageNet, OpenImages, and the proprietary dataset.

5.1 Experimental Setup

We followed the same general procedure for both active learning and search across all datasets and selection strate-

	Number of Concepts ($\|R\|$)	**Embedding Model (G_z)**	**Number of Examples ($\|U\|$)**	**Percentage Positive**
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%
10 billion (10B) images (proprietary)	8	ResNet-50 [He et al. 2016] (1000 classes)	10,094,719,767	-

Table 1: Summary of datasets

gies. Each experiment started with 5 positive examples because finding positive examples for rare concepts is challenging a priori. Negative examples were randomly selected at a ratio of 19 negative examples to every positive example to form a seed set L_r^0 with 5 positives and 95 negatives. The slightly higher number of negatives in the initial seed set improved average precision on the validation set across the datasets. The batch size b for each selection round was the same as the size of the initial seed set (i.e., 100 examples), and the max 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 concept 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 (dotted line) and standard deviation (shaded area around the line).

For similarity search, we used locality-sensitive hashing (LSH) [Charikar 2002] implemented in Faiss [Johnson, Douze, and Jégou 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, allowing experiments to run in parallel trivially. However, retrieval times for this approach were not as fast as Johnson, Douze, and Jégou [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 4 and differences in the "Selection" portion of the runtimes in Table 2. k was 100 for ImageNet and OpenImages unless specified otherwise, while the experiments on the 10 billion images dataset used a k of 1,000 or 10,000 to compensate for the size.

We split the data, selected concepts, and created embeddings as detailed below and summarized in Table 1. We also varied the embedding models, the value of k for SEALS, and the number of initial positives and negatives to test how robust SEALS was to our choices above. Across all values, SEALS performed similarly to the results presented here.

5.2 ImageNet

ImageNet [Russakovsky et al. 2015] has 1.28 million training images spread over 1,000 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 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.

Active learning. 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 2,000 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. The reduced skew from the initial seed set only accounted for a small part of the improvement, as Random-SEALS achieved an mAP of only 0.498.

Active search. As expected, MLP-All and MLP-SEALS significantly outperformed all 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.

5.3 OpenImages

OpenImages [Kuznetsova et al. 2020] has 7.34 million images from Flickr. However, only 6.82 million images were still available in the training set at the time of writing.

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

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 search index, SEALS substantially improved runtimes across datasets and strategies.

The human-verified labels provide partial coverage for over 19,958 classes. Like Kuznetsova et al. [2020], we treat examples that are not positively labeled for a given class as negative examples. This label noise makes the task much more challenging, but all of the selection strategies adjust after a few rounds. 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 predefined test split for evaluation.

Active learning. 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$\times$ and the total time by over 3$\times$, as shown in Table 2. Increasing k to 1,000 significantly narrowed this gap for MaxEnt-SEALS and MLP-SEALS, improving mAP to 0.395. Moreover, the reduced candidate pool from SEALS made ID tractable, whereas ID-All ran for over 24 hours in wall-clock time without completing a single round (Table 2).

Active search. The gap between the baselines and SEALS was even closer on OpenImages than on ImageNet 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.

5.4 10 Billion Images

10 billion (10B) publicly shared images from a large internet company were used to test SEALS' scalability. We used the same pre-trained ResNet-50 model as the OpenImages experiments. We also selected eight additional classes from OpenImages as rare concepts: 'rat,' 'sushi,' 'bowling,' 'beach,' 'hawk,' 'cupcake,' and 'crowd.' This allowed us to use the 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.

Active learning. Despite the 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 (Figure 1). This reduction allowed MaxEnt-SEALS to finish selection rounds in just seconds on a single 24-core machine, while MaxEnt-All took several minutes on a cluster with tens of thousands of cores. Unlike the ImageNet and OpenImages experiments, MLP-SEALS performed poorly at this scale because there are likely many redundant or near-duplicate examples of little value.

Active search. SEALS performed well despite considering less than 0.1% of the data and collected two orders of magnitude more positive examples than random sampling.

6 Discussion

Latent structure of unseen concepts. To better understand when SEALS works, we analyzed the relationship between average precision and the structure of the nearest

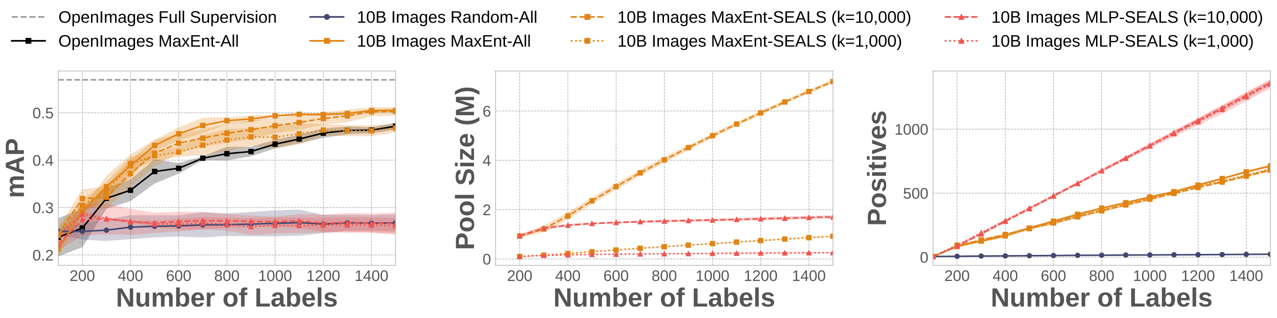
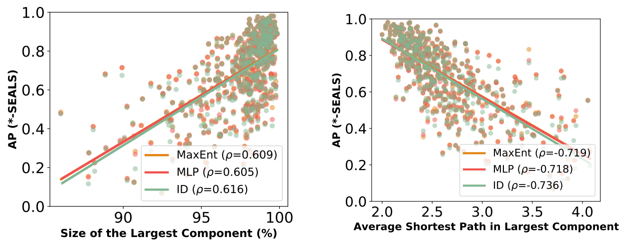
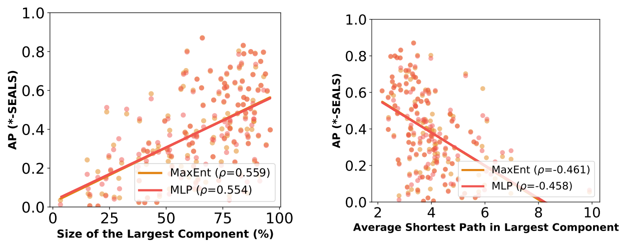


Figure 1: Active learning and search on a de-identified and aggregated dataset of 10 billion publicly shared images. Across strategies, SEALS with $k = 10,000$ performed similarly to the baseline approach in terms of both the error the model achieved for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the data U (middle).



(a) ImageNet



(b) OpenImages

Figure 2: Correlations between AP and measurements of the latent structure of unseen concepts. SEALS ($k = 100$) achieved higher APs for classes that formed larger connected components (left) and had shorter paths between examples (right) in ImageNet (top) and OpenImages (bottom).

neighbor graph across concepts. Overall, SEALS performed better for concepts that formed larger connected components and had shorter paths between examples, as shown in Figure 2. 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 5, 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 as in [Ertekin et al. 2007], mAP and recall would drop for all strategies. 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. On a per-class level, concepts like ‘monster truck’ and ‘blackberry’ performed much better than generic concepts like ‘electric blue’ and ‘meal’ that were more scattered. This fragmentation partly explains the gap between SEALS and the baselines in Section 5, 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, SEALS led to significant gains over random sampling. This makes active learning and search tractable for almost any amount of unlabeled data and computational budget, as demonstrated with the 10B images dataset.

7 Conclusion

We introduced SEALS as a simple approach to make the selection rounds of active learning scale sublinearly with the unlabeled data. Instead of scanning over all of the data, SEALS restricted the candidate pool to the nearest neighbors of the labeled set. Despite this limited pool, we found that SEALS achieved similar average precision and recall while improving computational efficiency by up to three orders of magnitude, enabling *web-scale active learning*.

8 Acknowledgments

This Stanford research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Ant Financial, Facebook, Google, and VMware—as well as Toyota Research Institute, Cisco, SAP, and the NSF under grants DGE-114747 and CNS-1651570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Toyota Research Institute (”TRI”) provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

Abu-El-Haija, S.; Kothari, N.; Lee, J.; Natsev, P.; Toderici, G.; Varadarajan, B.; and Vijayanarasimhan, S. 2016. YouTube-8M: A Large-Scale Video Classification Benchmark. arXiv:1609.08675.

Amodei, D.; and Hernandez, D. 2018. AI and Compute.

Ashmawy, K.; Yi, S.; and Chao, A. 2019. Searchable Ground Truth: Querying Uncommon Scenarios in Self-Driving Car Development. https://eng.uber.com/searchable-ground-truth-atg/.

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

Babenko, A.; Slesarev, A.; Chigorin, A.; and Lempitsky, V. 2014. Neural codes for image retrieval. In *European conference on computer vision*, 584–599. Springer.

Balcan, M.-F.; and Long, P. 2013. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, 288–316. PMLR.

Beluch, W. H.; Genewein, T.; Nürnberger, A.; and Köhler, J. M. 2018. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9368–9377.

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

Chelba, C.; Mikolov, T.; Schuster, M.; Ge, Q.; Brants, T.; Koehn, P.; and Robinson, T. 2013. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. Technical report, Google.

Coleman, C.; Yeh, C.; Mussmann, S.; Mirzasoleiman, B.; Bailis, P.; Liang, P.; Leskovec, J.; and Zaharia, M. 2020. Selection via Proxy: Efficient Data Selection for Deep Learning. In *International Conference on Learning Representations*.

Ertekin, S.; Huang, J.; Bottou, L.; and Giles, L. 2007. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 127–136.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep Bayesian Active Learning with Image Data. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1183–1192. PMLR.

Garnett, R.; Krishnamurthy, Y.; Xiong, X.; Schneider, J.; and Mann, R. 2012. Bayesian Optimal Active Search and Surveying. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, 843–850. Madison, WI, USA: Omnipress. ISBN 9781450312851.

He, J.; and Carbonell, J. G. 2007. Nearest-neighbor-based active learning for rare category detection.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*.

Holstein, K.; Wortman Vaughan, J.; Daumé III, H.; Dudik, M.; and Wallach, H. 2019. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–16.

Jain, P.; Vijayanarasimhan, S.; and Grauman, K. 2010. Hashing hyperplane queries to near points with applications to large-scale active learning. *Advances in Neural Information Processing Systems*, 23: 928–936.

Jiang, S.; Garnett, R.; and Moseley, B. 2019. Cost Effective Active Search. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 4880–4889. Curran Associates, Inc.

Jiang, S.; Malkomes, G.; Abbott, M.; Moseley, B.; and Garnett, R. 2018. Efficient nonmyopic batch active search. In *Advances in Neural Information Processing Systems*, 1099–1109.

Jiang, S.; Malkomes, G.; Converse, G.; Shofner, A.; Moseley, B.; and Garnett, R. 2017. Efficient Nonmyopic Active Search. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1714–1723. International Convention Centre, Sydney, Australia: PMLR.

Johnson, J.; Douze, M.; and Jégou, H. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734*.

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N. 2009. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2372–2379. IEEE.

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N. 2012. Coverage optimized active learning for k-nn classifiers. In *2012 IEEE International Conference on Robotics and Automation*, 5353–5358. IEEE.

Karpathy, A. 2018. TRAIN AI 2018 - Building the Software 2.0 Stack.

Karpathy, A. 2020. AI for Full-Self Driving.

Kirsch, A.; van Amersfoort, J.; and Gal, Y. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, 7024–7035.

Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; Duerig, T.; and Ferrari, V. 2020. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*.

Lee, K.; Rao, V.; and Arnold, W. C. 2019. Accelerating Facebook's infrastructure with application-specific hardware. https://engineering.fb.com/data-center-engineering/accelerating-infrastructure/.

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

Lin, C.; Mausam, M.; and Weld, D. 2018. Active Learning with Unbalanced Classes and Example-Generation Queries. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 6.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2537–2546.

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

Pinsler, R.; Gordon, J.; Nalisnick, E.; and Hernández-Lobato, J. M. 2019. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*, 6356–6367.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*.

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

Settles, B. 2011. From Theories to Queries: Active Learning in Practice. In Guyon, I.; Cawley, G.; Dror, G.; Lemaire, V.; and Statnikov, A., eds., *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, 1–18. Sardinia, Italy: PMLR.

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

Settles, B.; and Craven, M. 2008. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 1070–1079. USA: Association for Computational Linguistics.

Sinha, S.; Ebrahimi, S.; and Darrell, T. 2019. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5972–5981.

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. YFCC100M. *Communications of the ACM*, 59(2): 64–73.

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

Wan, M.; Misra, R.; Nakashole, N.; and McAuley, J. J. 2019. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In Korhonen, A.; Traum, D. R.; and Màrquez, L., 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*, 2605–2610. Association for Computational Linguistics.

Wang, Y.-X.; Ramanan, D.; and Hebert, M. 2017. Learning to model the tail. In *Advances in Neural Information Processing Systems*, 7029–7039.

Warmuth, M. K.; Liao, J.; Rätsch, G.; Mathieson, M.; Putta, S.; and Lemmen, C. 2003. Active learning with support vector machines in the drug discovery process. *Journal of chemical information and computer sciences*, 43(2): 667–673.

Warmuth, M. K.; Rätsch, G.; Mathieson, M.; Liao, J.; and Lemmen, C. 2002. Active learning in the drug discovery process. In *Advances in Neural information processing systems*, 1449–1456.

Wei, K.; Iyer, R.; and Bilmes, J. 2015. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, 1954–1963.

Yoo, D.; and Kweon, I. S. 2019. Learning loss for active learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 93–102.

Zhang, B.; Li, L.; Yang, S.; Wang, S.; Zha, Z.-J.; and Huang, Q. 2020. State-relabeling adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8756–8765.

Zhang, X.; Fang, Z.; Wen, Y.; Li, Z.; and Qiao, Y. 2017. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, 5409–5418.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 649–657.

Zhu, J.-J.; and Bento, J. 2017. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*.