ACTIVE LEARNING AT THE IMAGENET SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

Active learning (AL) algorithms identify an optimal subset of data for annotation, such that deep neural networks (DNN) can achieve better performance when trained on this labeled subset. AL is especially impactful in industrial scale settings where data labeling costs are high and practitioners use every available tool to improve model performance. The recent success of self-supervised pretraining (SSP) highlights the importance of harnessing abundant unlabeled data to boost model performance. By combining AL with SSP, we make use of unlabeled data while simultaneously labeling and training on particularly informative samples. We study a combination of AL and SSP on ImageNet. We find that performance on small toy datasets – the typical benchmark setting in the literature – is not representative of performance on ImageNet due to the class imbalanced samples selected by an active learner. Among the existing baselines we test, popular AL algorithms across a variety of small and large scale settings fail to outperform random sampling. To remedy the class-imbalance problem, we propose Balanced Selection (BASE), a simple, scalable AL algorithm that outperforms random sampling consistently by selecting class balanced samples for annotation.

1 INTRODUCTION

Fueled by the success of deep learning, the global data annotation market is projected to reach $3.4 billion by 2028 Research & Markets (2021). The data labeling process is a daunting hurdle for institutions aiming to deploy deep learning models at an industrial scale. The annotation process is slow, costly, and in some cases requires domain-expert annotators.

A large body of machine learning research seeks to reduce data labeling costs by harnessing as much information as possible directly from unlabeled data or by leveraging other labeled datasets whenever possible. Ultimately, however, labeled data is required to achieve adequate deep learning model performance, especially in mission critical scenarios. Due to time and budget constraints, practitioners are often restricted to selecting a small subset of the available data for annotation. This restriction raises the following question: What is the best approach for selecting this subset?

Active learning (AL) is a subfield of machine learning (ML) dedicated to answering this question. Given a large pool of unlabeled data and a fixed labeling budget, an AL algorithm selects a subset of the unlabeled data to be annotated. Once labeled, the subset is subsequently used to train a ML model. The goal of the active learner is to select the subset that will optimize the generalization performance of the ML model.

In this work, we focus on classification tasks using deep neural networks (DNN). We study a combination of AL and self-supervised pretraining (SSP) in the large data regime. Large-scale data is prevalent in real-world scenarios, where unlabeled data is typically abundant and cheap to collect. Furthermore, in real-world settings, practitioners are compelled to leverage the available unlabeled data in order to achieve adequate model performance at the lowest possible annotation cost. State-of-the-art SSP methods can provide these performance boosts at no annotation cost.

Prior research on AL focuses on the CIFAR-10, CIFAR-100, and SVHN Krizhevsky (2009); Yang & and (2019) datasets to compare AL algorithms. However, it is unclear whether performance on these datasets is predictive of performance on real-world datasets that are orders of magnitude larger and that may contain many more classes or even imbalanced data. We focus particularly on ImageNet Russakovsky et al. (2015), as it contains 1000 classes, 1.2 million images, and a significant amount of label noise Beyer et al. (2020); Stock & Cissé (2018); Tsipras et al. (2020) as is common in industrial

settings Liao et al. (2021). AL cost savings are much more impactful at the ImageNet scale and beyond, and cannot be understood by studying small datasets alone. These cost savings are due in part to the sheer amount of available data but also the ambiguity of the classes considered. To curate ImageNet, each image was presented to multiple human annotators who voted until a consensus was reached on the label Deng et al. (2009). This voting mechanism translates directly to high annotation costs.

Finally, we specifically focus on the interaction of AL with SSP. SSP has been shown to provide a significantly larger accuracy boost compared to AL alone Siméoni et al. (2021). It is therefore important to study whether AL offers any additional benefits on top of SSP. We present the first AL results on ImageNet using SSP.

We summarize our contributions below:

1. We demonstrate that the performance of popular AL methods, which has been observed on small datasets, does not transfer to the larger and more complex ImageNet challenge. In fact, on the common linear evaluation task (see Section 3), most popular AL algorithms perform worse than random sampling on ImageNet.
2. We identify and study sampling imbalance as a major failure mode for AL algorithms. Because ImageNet has many classes with highly heterogeneous properties, AL algorithms have a tendency to heavily sample from preferred classes while nearly ignoring others. This problem is less severe on simple tasks with fewer and more homogeneous classes.
3. We introduce the Balanced Selection (BASE) AL strategy. BASE efficiently selects images that lie near class boundaries in feature space while also promoting an even class distribution. By carefully selecting which data to label, BASE achieves significantly better sample efficiency than standard self-supervised learning (SSL) pipelines that rely on random sampling.
4. We show, for the first time, that AL can offer performance boosts on ImageNet when combined with SSL. Our BASE algorithm, when used to train a classifier on top of a SSL feature extractor, matches the top-5 accuracy of the state-of-the-art EsViT Li et al. (2021) SSL algorithm while using only 55% of the labels.

We make our code publicly available in the hopes that others can easily reproduce our results and use our codebase for future AL research [1].

2 BACKGROUND & RELATED WORK

A typical AL algorithm cycles between learning from a small amount of labeled data, using the model to gather information about the unseen unlabeled data, and using this information to choose a subset of the unlabeled data to be manually annotated. This cycle then repeats, with the labeled dataset increasing in size at every iteration, until a pre-determined manual-annotation budget is exhausted. In this section, we provide a formal description of the AL problem, followed by an overview of existing methods.

We will adopt the notation from Sener & Savarese (2018) with slight modifications to accommodate more general cases. We will study a C-way classification problem defined over a compact space $\mathcal{X} = \mathbb{R}^d$ and a finite label set $\mathcal{Y} = \{1, \ldots, C\}$. We denote the loss function by $l(\cdot, \cdot, \mathbf{w}) : \mathcal{X}, \mathcal{Y} \to \mathbb{R}$, where $\mathbf{w}$ are the parameters (i.e., the weights) of a classifier $f(\mathbf{w}, \cdot) : \mathcal{X} \to \mathcal{Y}$, which we simply denote as $f(x)$ in the rest of the paper.

The entire dataset is a collection of n points $Z \subseteq \mathcal{X} \times \mathcal{Y}$ sampled *i.i.d.* over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ as $\{x_i, y_i\}_{i \in [n]} \sim p_{\mathcal{Z}}$. Initially, some subset of m points is assumed to have been annotated by an expert, we will denote the indices of those points by $s^0 = \{s^0(j) \in [n]\}_{j \in [m]}$.

An AL algorithm has access to $\{x_i\}_{i \in n} \subseteq \mathcal{X}$ but only the labels with indices s^0, i.e. $\{y_{s^0(j)}\}_{j \in [m]}$. The algorithm is also given a budget b of queries to ask an oracle (typically a human annotator), and a learning algorithm A_s which outputs a set of parameters $\mathbf{w}$ given $\{x_i\}_{i \in n} \subseteq \mathcal{X}$ and $\{y_{s(j)}\}_{j \in [m]}$. The goal of AL is to identify a new subset s^1 of unlabeled data such that:

$$s^1 = \underset{s^1 : |s^1| < b}{\arg\min} \, \mathbb{E}_{x, y \sim p_{\mathcal{Z}}} \left[l(x, y; A_{s^0 \cup s^1}) \right]. \tag{1}$$

---

[1] https://anonymous.4open.science/r/Active-Learning-At-The-ImageNet-Scale/README.md

The above formulation constitutes one round of active learning. Typically, the algorithm runs for K rounds producing a sequence of subsets $s^1, s^2 \ldots, s^K$ to be labeled by the oracle and added to the labeled dataset for the following round. We will denote by D_L^k all the indices of points labeled before the start of round k, i.e., $D_L^k = \bigcup_{i=0}^{k-1} s_i$, and likewise $D_U^k = [n] \setminus D_L^k$ is the set of all unlabeled points at round k. We will use $| \cdot |$ to refer to the cardinality of a set.

Selection Methodologies Existing AL algorithms for DNNs can be roughly broken down into two categories: those designed to tackle class imbalanced datasets and those that are not. The wide majority of existing algorithms fall in the latter category.

Algorithms designed for balanced datasets can be further broken down into two categories: uncertainty based sampling and density based samplingAggarwal et al. (2020). Uncertainty based AL algorithms operate by first quantifying the classifier's uncertainty about its prediction on every unlabeled sample at round k Gal & Ghahramani (2016); Gal et al. (2017); Settles (2010), then querying the examples on which the classifier is deemed most uncertain. Prediction entropy, least confidence, and margin sampling are commonly used uncertainty measures. Intuitively, uncertainty based sampling improves the model's prediction on subsets of the domain $mathcalX$ where the model is most uncertain, and in turn, improves the model's generalization ability.

On the other hand, density based algorithms Sener & Savarese (2018); Ash et al. (2020) generate high-dimensional features from the data then select examples with the most representative features. In practice, this selection involves running a clustering algorithm Ash et al. (2020); Citovsky et al. (2021) or finding coresets Sener & Savarese (2018) in feature space. The features can be obtained by removing the network's linear classification head Sener & Savarese (2018); Citovsky et al. (2021) or by taking its gradients with respect to every sample Ash et al. (2020). Intuitively, density based sampling ensures that the most densely populated regions of space, which contain the most data at test time, are represented in the labeled set.

A smaller minority of AL algorithms specifically tackle class imbalanced data Aggarwal et al. (2020). However, as we discuss below, techniques in this area cannot scale to ImageNet.

Scaling Ability AL strategies designed for classical machine learning (ML) focus on querying a single label (i.e. $b = 1$), re-training the model on all labeled data, querying the next example, retraining again, etc. This has obvious advantages as the active learner is given access to the current label and can therefore use it to guide its selection strategy. However, as datasets increased in size, ML algorithms became costly to train, and data annotation grew into an entire industry. It is now necessary for AL algorithms to operate at a large scale.

To be practical for neural network applications, AL strategies must be able to query a large batch of data at once, receive labels for the entire batch, then query another batch (i.e. $b >> 1$). This batch AL approach minimizes costs and time by keeping a large group of annotators occupied and by reducing the number of training runs needed to update DNNs on newly acquired data. At the same time, modern AL strategies must be able to efficiently process a massive pool of unlabeled data at each round (i.e. large D_U^k).

Table 1. Performance gains offered by VAAL Sinha et al. (2019) on ImageNet vs those offered by MoCo v2 Chen et al. (2020b), a popular SSP method. SSP, when combined with random sampling, yields a substantially larger boost in performance.

% of labels	SSP	Strategy	Accuracy
	No	Random	50%
25%	No	VAAL	+1.5%
	Yes	Random	+16.65%

In Table 4 in Appendix and Section 5, we discuss the time complexities of several baselines considered in this paper in greater detail.

Usage of Unlabeled Data Unlabeled data is often abundant in real-world scenarios, and leveraging it effectively can lead to significant reductions in data annotation costs. In Chan et al. (2021); Siméoni et al. (2021), the authors study the benefits of AL when combined with both SSP and FixMatch, a semi-supervised technique.

In this work, we restrict our experimental setup to SSP because semi-supervised techniques would require training to saturation multiple times on the entire Imagenet dataset Sohn et al. (2020), which is prohibitively expensive. In Table 1, we show that using only 25% of the ImageNet labels, the MoCo v2 SSP method with randomly sampled labels is able to boost model performance by 18.15 percentage points, whereas carefully selecting examples using the VAAL Sinha et al. (2019) AL algorithm only boosts performance by 1.5 percentage points if the classifier's weights are randomly initialized. Clearly, initializing models with SSP offers strong advantages in the label scarce regime. In section 5, we show for the first time that AL offers additional performance gains on top of SSP at the ImageNet scale. To the best of our knowledge, only two prior works Beluch et al. (2018); Sinha et al. (2019) study AL on ImageNet, and none study the interaction between AL and SSP at this scale.

3 LINEAR EVALUATION TASK

When DNNs are deployed at an industrial scale, it is common to use a shared backbone network for feature extraction, then apply separate heads directly on the features to accomplish different downstream tasks. A single shared backbone is easier to maintain and carries a small memory footprint, making it more practical for edge devices. Training this entire pipeline end-to-end is time consuming and computationally intensive. Therefore, in this setting, as new labeled data becomes available, the different task-specific heads are frequently finetuned while keeping the backbone frozen. The backbone itself is updated less frequently[2]. In fact, with the emergence of massive foundation models Bommasani et al. (2021), such as GPT-3 Brown et al. (2020), BERT Devlin et al. (2019), and DALL-E Ramesh et al. (2021), practitioners may only have restricted access to the backbone. Consequently, it is important to evaluate AL algorithms when the feature extractor is fixed and only the classification head is finetuned. This task is a common benchmark in self-supervised learning (SSL) research Li et al. (2021); Chen et al. (2020a;b), where the proposed SSL method is used to pretrain the network's feature extractor, then a linear classification head is trained on the features in a fully-supervised fashion. However, to the best of our knowledge, AL strategies on ImageNet have not been evaluated in this specific setting.

4 METHODS

We now describe our proposed method. We start with a simple preliminary variant of uncertainty based selection. We then show how this variant can be adapted to prevent sampling imbalances from emerging and accumulating over rounds, resulting in improved performance.

Margin Selection We first introduce a simplified variant of our Balanced Selection algorithm, which we call Margin Selection (MASE). MASE selects the b examples closest to any decision boundary at every round of AL. Intuitively, these samples should have the most influence on the decision of the model. We define distance to decision boundary (DDB) as follows,

$$\text{DDB}(x) = \min_\epsilon ||\epsilon||_2 \quad \text{s.t.} f(x + \epsilon) \neq f(x). \quad (2)$$

Algorithm 1 Algorithm for MASE

Input: Query Budget b, Indices of Labeled Samples D_L^k
$s \leftarrow D_L^k$
while $|s| < |D_L^k| + b$ **do**
  $x^* \leftarrow \arg\min_{x \in [N] \setminus s} \text{DDB}(f(x))$
  $s \leftarrow s \cup \{x^*\}$
end while
return $s \setminus D_L^k$

When f is a DNN, DDB is expensive to compute in input space. We instead estimate this distance in feature space. For the models considered in this paper, the final layer is a linear classification head on top of the features produced by a feature extractor; therefore, computing DDB in feature space reduces to computing the projection of the feature vector onto the normal vector of the linear decision boundary, which can be implemented very efficiently. We provide pseudocode for MASE in Algorithm 1.

To the best of our knowledge, MASE is a novel AL strategy, similar algorithms were only studied on 2-class classification tasks using support vector machines Ertekin et al. (2007), or a different definition of distance Cho et al. (2021).

---

[2]The quintessential manifestation of this framework is described in Tesla's AI day: https://youtu.be/j0z4FweCy4M?t=3300 (55th minute).

Algorithm 2 Algorithm for BASE

Input: Query Budget b, Number of Classes C, Indices of Labeled Samples D_L^k
$s \leftarrow D_L^k$
while $|s| < |D_L^k| + b$ **do**
   for c in $[C]$ **do**
      $x^* \leftarrow \arg\min_{x \in [N] \setminus s} \text{DCSDB}(f(x)), c)$
      $s \leftarrow s \cup \{x^*\}$
   end for
end while
return $s \setminus D_L^k$

Balanced Selection In Section 5, we show that only a single baseline AL algorithms beats random sampling on the linear evaluation task on ImageNet and only by a relatively small margin. This is partially due to the imbalance induced by the active learner, which does not query examples evenly across classes. Motivated by this observation, we design Balanced Selection (BASE), an AL strategy capable of scaling efficiently while also querying a balanced batch of examples. As opposed to naïvely choosing the examples with the smallest DDB, BASE selects examples based on their distance to class specific decision boundaries (DCSDB), defined as

$$\text{DCSDB}(x, c) = \begin{cases} \min_\epsilon \|\epsilon\|_2 & \text{s.t. } f(x + \epsilon) = c \text{ if } f(x) \neq c \\ \min_\epsilon \|\epsilon\|_2 & \text{s.t. } f(x + \epsilon) \neq c \text{ if } f(x) = c. \end{cases}$$

More specifically, for each class $c \in \{1, \ldots, C\}$, BASE selects the b/C samples with the smallest $\text{DCSDB}(x, c)$. Similar to our MASE implementation, we only consider distances in feature space. We provide a visual illustration of BASE's selection strategy in Fig. 1 and its pseudocode in Algorithm 2. The time complexity of Algorithm 2 is dominated by computing DCSDBs, but those can be computed once and stored, so the algorithm runs in $\mathcal{O}(C \cdot (d' + \log(b)) \cdot |D_U^k|)$ in practice, where d' is the dimension of the features. This puts BASE on par with the fastest baselines – see 4.
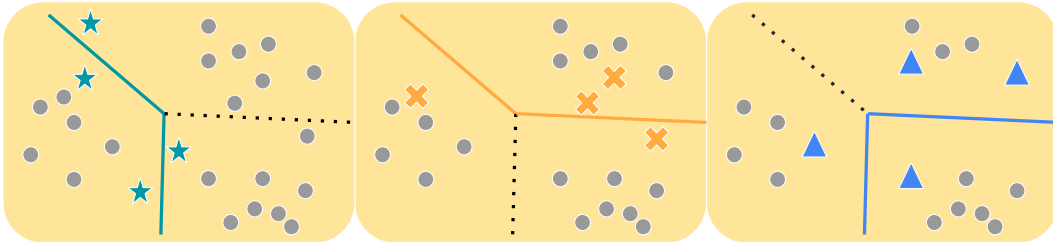

Figure 1. An illustration of BASE for 2-dimensional features and a 3 class problem. The algorithm selects an equal number of points (shown using colored stars, crosses, and triangles) that are closest to each decision boundary (solid lines).

5 EXPERIMENTS

In this section, we outline our experimental design, followed by a presentation of our results. In figure captions, we will refer to different experiments using a capital letter for the dataset/model combination, and a capital roman numeral for the experimental setup. For example, setting A-I refers to AL strategies tested on CIFAR-10 using a ResNet-18 with the model weights initialized using SSP at every round. Baseline strategies are referenced using lower case roman numerals, e.g., v refers to BADGE. Below, we enumerate each dataset/model combination, training setup, and AL method, and assign each a letter or numeral. All other implementation details can be found in Appendix A.

Datasets and Models In our experiments, we use the following dataset and model architecture combinations.

A. CIFAR-10 Krizhevsky (2009) w/ ResNet-18 He et al. (2016).

B. Imbalanced CIFAR-10 Krizhevsky (2009) w/ ResNet-18 He et al. (2016): The number of samples per class decreases exponentially from the most frequent class to the least frequent class; the most sampled class contains $10\times$ the number of samples in the least sampled class Cao et al. (2019a).

C. ImageNet Russakovsky et al. (2015) w/ ResNet-50 He et al. (2016).

D. ImageNet Russakovsky et al. (2015) w/ ViT Dosovitskiy et al. (2021).

Training Setups   We consider two different settings for training the classifier.

I. **End-to-end finetuning from a self-supervised checkpoint.** We first train the network using self-supervised learning on all available unlabeled data. At every round k of AL, the backbone's weights (all layers except the final linear classifier) are reset to the SSP weights then the network is finetuned end-to-end on all the available labeled data D_L^k.

II. **Linear evaluation from a self-supervised checkpoint.** Here, we employ SSP. At every AL round k, we use the SSP checkpoint, but we only update the final linear layer of the network on labeled data D_L^k.

Baselines   We compare BASE (ours) to the following baselines.

i. Random Sampler. Queries samples from D_U^k uniformly at random.

ii. Balanced Random Sampler. Iterates over classes and chooses an equal number of examples uniformly at random from each class. *This baseline strategy cheats, as it requires the labels for points in D_U^k in order to make its selection. We include it only for scientific purposes.*

iii. Coreset AL. Sener & Savarese (2018) We solve the k-center problem using the classical greedy 2-approximation, as described in Algorithm 1 of Sener & Savarese (2018).

iv. Partitioned Coreset Sampler. Citovsky et al. (2021) Partitions the dataset into p partitions, then runs the coreset algorithm to select b/p examples from each partition. This implementation only calculates pairwise distances on smaller subsets, which is more computationally efficient.

v. BADGE AL Ash et al. (2020). Calculates the gradient with respect to the last linear layer, then applies the K-means++ seeding algorithm Arthur & Vassilvitskii (2007) on the gradients. On ImageNet, the size of the gradient embedding is proportional to the number of classes, which makes it $100\times$ larger than CIFAR-10 embeddings. Furthermore, the K-means++ seeding algorithm requires the pairwise distances, which again is computationally prohibitive on ImageNet.

vi. Partitioned BADGE Sampler. Citovsky et al. (2021) BADGE with a similar partitioning trick as Partitioned Coreset, and global pooling to reduce the embedding dimension.

vii. Confidence Sampler. Selects the examples with the smallest top logit (least confidence).

viii. Margin SamplerScheffer et al. (2001). Selects examples with the smallest differences between the top logit and the second largest logit (minimum margin).

ix. VAAL Sinha et al. (2019). Trains a binary classifier to distinguish between features produced by labeled vs unlabeled samples. The features are obtained by training a variational autoencoder. Queries the unlabeled samples that the binary classifier is most confident about.

x. Balancing Sampler Aggarwal et al. (2020) Calculates cluster centers for each class in feature space, then targets the class with the least number of queried examples, and finally selects examples that are close to the target class' center and away from other clusters.

xi. MASE (ours). See Section 4.

Scalability of baseline methods   Coreset (iii) and BADGE (v) are prohibitively expensive to run at the ImageNet scale – see Table 4. Additionally, both algorithms require storing large tensors in memory – $O(d' \cdot |D_U^k|)$ space complexity – as they require solving an optimization problem in feature space. For this reason, we exclude them from our comparisons, and instead implement Partitioned Coreset (iv) and Partitioned BADGE (vi), two scalable variants of the original strategies Citovsky et al. (2021). We also note that the Balancing Sampler (x) is not a batch AL algorithm as it acquires labels one at a time, making it impractical in terms of both computation and human labeling bandwidth in large-scale settings.

5.1 Solving Class Imbalance Allows Scaling

This section will show that class balanced sampling is important to scaling. Without explicitly imposing balance, imbalance becomes a problem and causes baseline algorithms to under-perform
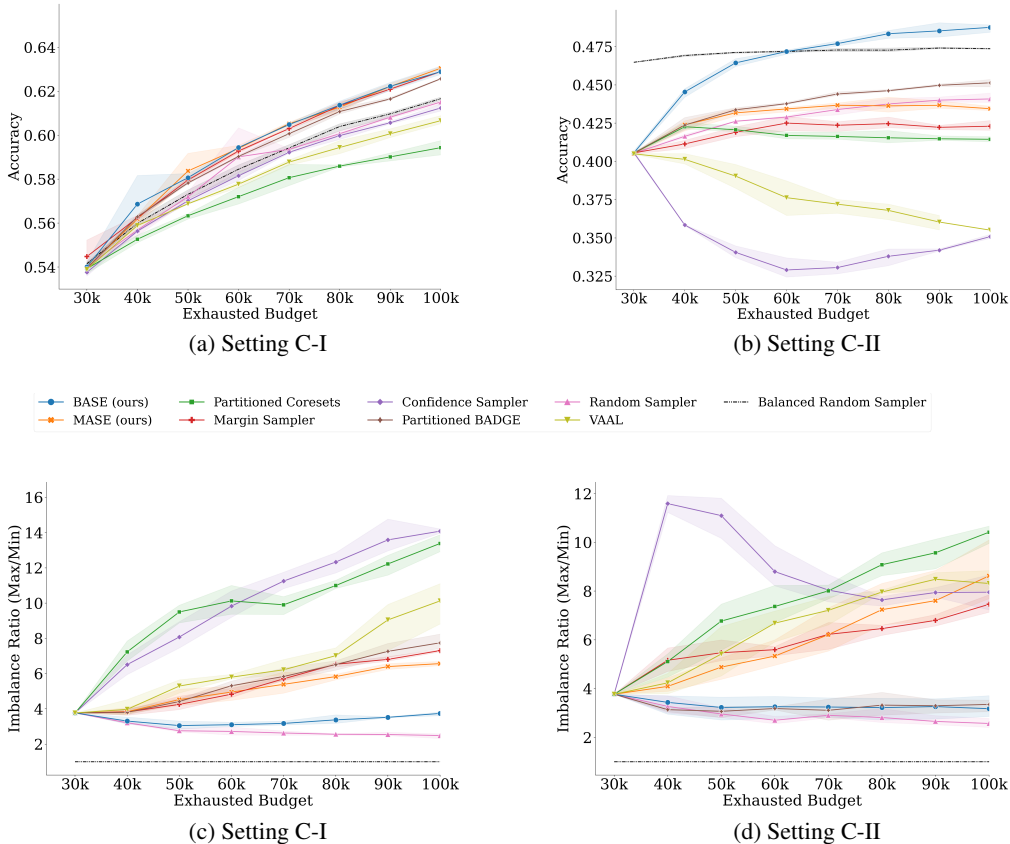
Figure 2. Average ImageNet accuracy and imbalance ratio over 3 runs. Shaded regions depict the 95% confidence interval of the results. Figures 2a and 2c are obtained by training a ResNet-50 end-to-end starting from a SSP checkpoint at every AL round. In contrast, Figures 2b and 2d are obtained by finetuning only the final linear layer at every round.

random sampling at the ImageNet scale. But by querying balanced samples, our BASE algorithm recovers the good properties of AL in the large-scale regime, and even matches the state-of-the-art EvSiT Li et al. (2021) results using only 71% of the ImageNet labels.

5.1.1 BASELINES PERFORM POORLY ON IMAGENET

Here, we analyze the performance of all baselines on ImageNet. In Figure 2a, we compare different baselines on ImageNet, starting from a SSP checkpoint and finetuning the network end-to-end at each AL round. Three baselines provide material performance boosts over random sampling in that setting: Margin Sampler (viii), Partitioned BADGE (vi), and our MASE algorithm (xi). In Figure 2b, we evaluate all baselines on ImageNet in the linear evaluation setting described in Section 3. Surprisingly, *only a single baseline outperforms random sampling on the linear evaluation task.*

5.1.2 THE IMPORTANCE OF BALANCED SAMPLING

Figures 2c and 2d compare *class imbalance ratios* – the number of labels from the most sampled class over that of the least sampled class – for each sampling strategy. Most baseline samplers disproportionately query certain classes. Indeed, the Confidence Sampler, the worst performing baseline in Figure 2b, induces an imbalance ratio close to 12 after the first round of AL. To further investigate the effects of class imbalance, we implement a cheating baseline strategy (Balanced Random Sampler ii) which queries a perfectly balanced batch at each round. On the end-to-end finetuning experiment in Figure 2a, querying balanced batches is not sufficient to outperform random sampling. However, on the linear evaluation task in Figure 2b, the cheating Balanced Random Sampler (ii) outperforms random sampling by approximately 6 percentage points at every round.

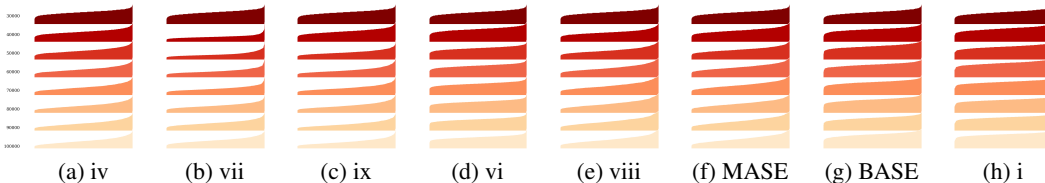| (a) iv | (b) vii | (c) ix | (d) vi | (e) viii | (f) MASE | (g) BASE | (h) i |

Figure 3. Setting C-II. The distribution of D_U^k at every AL round for different strategies on ImageNet in the linear evaluation setting. All experiments start with the same randomly selected subset s^0. The x-axis is sorted for each histogram (every row in every subplot) from least queried class to most queried class. The height of the histogram at a given location on the x-axis indicates the proportion of the examples sampled from that class. BASE is visibly the most balanced strategy after random sampling.

Table 2. When applying BASE on the linear evaluation task with the state-of-the-art EsViT Li et al. (2021) SSP method, we are able to achieve the same state-of-the-art linear evaluation accuracy on ImageNet with only 71% (for top-1 acc.) and 55% (for top-5 acc.) of the data. The bolded number indicates the smallest amount of required data to achieve the same accuracy as using all of the data for training.

		Number of Labels (% of all labels)					
	Strategy	0.3M (24%)	0.5M (39%)	0.7M (55%)	0.9M (71%)	1.1M (87%)	All(100%)
Top 1 Acc.	BASE (ours)	78.9%	80.5%	81.0%	**81.2%**	81.2%	81.2%
	Random	78.9%	79.9%	80.5%	80.7%	81.0%	**81.2%**
Top 5 Acc.	BASE (ours)	94.4%	95.2%	**95.5%**	95.5%	95.5%	95.5%
	Random	94.5%	94.9%	95.1%	95.3%	**95.5%**	95.5%

Class imbalance ratios do not fully describe the class imbalance across all classes, but only the extremes. To further investigate class imbalance, we analyze the distributions of D_L^k for all baselines on the ImageNet linear evaluation task in Figure 3. It is clear from the figure that all baselines exhibit long tailed distributions, and increase the imbalance over time. In Appendix B, we include yet another measure of class imbalance using entropy.

5.1.3 BASE OUTPERFORMS BASELINES AND MITIGATES CLASS IMBALANCE

In Section 4, we proposed BASE, an AL algorithm specifically designed to query class balanced data. BASE can significantly outperform random sampling on the ImageNet linear evaluation task shown in Figure 2b. In fact, BASE can even outperform the unrealistic Balanced Random Sampler (ii), which cheats by using knowledge of ground truth labels to achieve perfect class balance. On the finetuning experiment in Figure 2a, BASE performs on par with the best two baselines in terms of accuracy. However, in Figures 3, 2c, and 2d, we show that our AL algorithm consistently achieves a more uniform class distribution than all other baselines on both the linear evaluation and end-to-end finetuning ImageNet tasks. The class distribution histograms for end-to-end finetuning can be found in Appendix C.

Finally, in Table 2, we show that by carefully selecting examples, BASE can reproduce the state-of-the-art linear evaluation top-1 accuracy results reported in EsViT Li et al. (2021) using $\sim 29\%$ less labeled data; BASE only needs 55% of the labels to match the same top-5 accuracy reported in Li et al. (2021).

5.2 CLASS IMBALANCE ON SMALL DATASETS

Motivated by our observations concerning the importance of balanced sampling in large-scale settings, we also investigate whether the balancing aspect of BASE offers benefits in small-scale settings. To this end, in Figure 4, we compare all AL strategies on an imbalanced version of CIFAR-10 Cao et al. (2019b) with an imbalance ratio of 10 starting from a SSP checkpoint obtained by training on the full imbalanced dataset. In this experiment, when training the classifier, we weigh each class differently in the loss function to penalize rare classes more heavily. The plots show a strong correlation between class distributions and performance of the AL algorithm. Balanced Random Sampler (ii), the "cheating" algorithm, achieves the best accuracy across the board. And with minor exceptions, for each algorithm, the better the performance in terms of accuracy, the less severe the observed class imbalance.

BASE is the best performing strategy in terms of accuracy and only second best in terms of class imbalance – the best being the non-scalable Balancing Sampler (x).
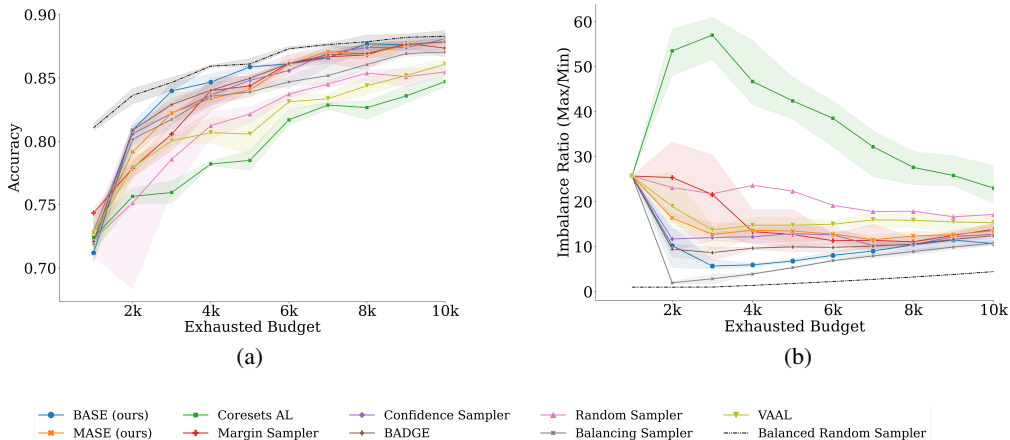
(a)

(b)

| BASE (ours) | Coresets AL | Confidence Sampler | Random Sampler | VAAL |
| MASE (ours) | Margin Sampler | BADGE | Balancing Sampler | Balanced Random Sampler |

Figure 4. Setting B-I with $|s^0| = b = 1000$. Average results over 3 runs on imbalanced CIFAR-10 obtained by finetuning a ResNet-18 end-to-end starting from a SSP checkpoint at every AL round. Shaded regions depict the 95% confidence interval of the results.

5.3 AL PERFORMS DIFFERENTLY WITH SSP

Throughout this paper, we argue that applying SSP along with simple random sampling is much more powerful than applying AL alone – see Table 1. Therefore, AL algorithms must prove that they can outperform random sampling in the SSP setting, otherwise they are redundant and potentially harmful to performance. In Figure 5, we show that some popular baselines, notably, Coreset AL (iii), are indeed harmful. A potential explanation for this failure mode can be found in Anonymous (2022), where the authors show that warm-starting the network weights at each round can negatively impact the performance of Coreset AL (iii). Warm-starting means to continue training the network starting with the weights obtained in the previous AL round, as opposed to randomly re-initializing the network weights at every round (cold start). We suspect that SSP, just like warm-starting, may negatively impact the performance of Coreset AL on CIFAR-10 and conclude that future research should not draw conclusions about the performance of an AL algorithm in the SSP setting solely by observing its behaviour in the cold starting setting.
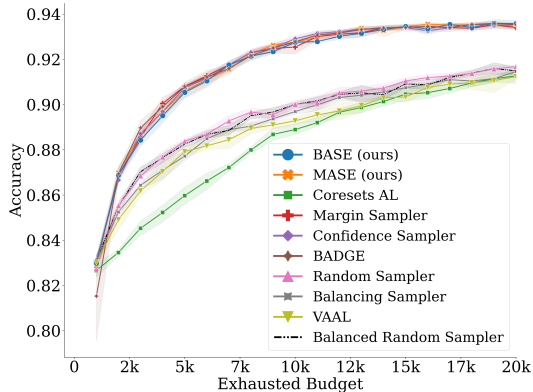

Figure 5. Setting A-I with $|s^0| = b = 1000$. Average results over 5 runs on CIFAR-10 obtained by training a ResNet-18 end-to-end starting from a SSP checkpoint at every AL round. Shaded regions depict the 95% confidence interval of the results. The 5 overlapping curves at the top are BASE, MASE, BADGE, Confidence Sampler, and Margin Sampler.

6 CONCLUSION

AL for DNNs is a very difficult problem to study, partly because we still cannot answer very fundamental questions about the generalization abilities of DNNs Huang et al. (2020), but also because *random sampling is an incredibly robust baseline*. In this paper, we highlighted the importance of stress-testing AL algorithms where they are most useful, namely on large-scale tasks. We showed that popular existing works cannot compete with random sampling across all settings, and we designed BASE, a robust AL strategy capable of doing just that. In future work, we hope to tackle more complex problems, where the cost savings incurred by AL are even more dramatic, such as large-scale segmentation and detection tasks.

REFERENCES

Umang Aggarwal, Adrian Popescu, and Céline Hudelot. Active learning for imbalanced datasets. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1417–1426, 2020. doi: 10.1109/WACV45572.2020.9093475.

Anonymous. Best practices in pool-based active learning for image classification. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7Rnf1F7rQhR. under review.

David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA '07*, 2007.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryghZJBKPS.

Nathan Beck, Durga Sivasubramanian, Apurva Dani, Ganesh Ramakrishnan, and Rishabh K. Iyer. Effective evaluation of deep active learning on image classification tasks. *CoRR*, abs/2106.15324, 2021. URL https://arxiv.org/abs/2106.15324.

William H. Beluch, Tim Genewein, Andreas Nurnberger, and Jan M. Kohler. The power of ensembles for active learning in image classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9368–9377, 2018. doi: 10.1109/CVPR.2018.00976.

Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet?, 2020.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *NIPS*, pp. 1565–1576, 2019a. URL https://proceedings.neurips.cc/paper/2019/hash/621461af90cadfdaf0e8d4cc25129f91-Abstract.html.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss, 2019b.

Yao-Chun Chan, Mingchen Li, and Samet Oymak. On the marginal benefit of active learning: Does self-supervision eat its cake? In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3455–3459. IEEE, 2021.

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

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Seong Jin Cho, Gwangsu Kim, and Chang D. Yoo. Least probable disagreement region for active learning, 2021. URL https://openreview.net/forum?id=bGPNpnZYr1.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. *CoRR*, abs/2107.14263, 2021. URL https://arxiv.org/abs/2107.14263.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: Active learning in imbalanced data classification. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pp. 127–136, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595938039. doi: 10.1145/1321440.1321461. URL https://doi.org/10.1145/1321440.1321461.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1050–1059. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045502.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning (ICML) - Volume 70*, ICML'17, pp. 1183–1192. JMLR.org, 2017.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016. doi: 10.1109/CVPR.2016.90.

W. Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K. Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

Yuan-Hong Liao, Amlan Kar, and Sanja Fidler. Towards good practices for efficiently annotating large-scale image classification datasets. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4348–4357, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

Research and Markets. Global data annotation tools market size, share & trends analysis report by type (text, image/video, audio), by annotation type (manual, automatic, semi-supervised), by vertical, by region, and segment forecasts, 2021-2028. Technical report, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher, and Gabriela Guimaraes (eds.), *Advances in Intelligent Data Analysis*, pp. 309–318, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

Burr Settles. Active learning literature survey. 07 2010.

Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. Rethinking deep active learning: Using unlabeled data at model training. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 1220–1227. IEEE, 2021.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. *CoRR*, abs/1904.00370, 2019. URL http://arxiv.org/abs/1904.00370.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Pierre Stock and Moustapha Cissé. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *ECCV*, 2018.

Ilya O. Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein autoencoders. *CoRR*, abs/1711.01558, 2017. URL http://arxiv.org/abs/1711.01558.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *ICML*, 2020.

Haoqi Yang and Hongge Yao and. Street view house number identification based on deep learning. *International Journal of Advanced Network, Monitoring and Controls*, 4(2470-8038):47–52, 2019. doi: 10.21307/ijanmc-2019-058.

A ADDITIONAL EXPERIMENTAL DETAILS

We provide additional details of implementations and hyperparameters in the following sections.

A.1 DATASET DETAILS AND EARLY STOPPING

Dataset Division We split the target dataset into training set, validation set and test set. All AL algorithms are restricted to query from the training set, and the initial pool is also sampled from training set. The validation set is used for early stopping.

CIFAR-10 comes with natural split of training and testing data. We keep the testing data as test set. We randomly sample 1% of the training data as validation set and keep the rest as training set.

For Imbalanced CIFAR-10, we keep the original testing split as test data. We then follow this implementation to subsample a set of long-tailed imbalance data from the training split. The set of imbalance data is then randomly partitioned into training/validation data with $0.99/0.01$ split ratio.

For ImageNet, the dataset itself comes with natural training split and validation split. We use the validation split as test set. We randomly sample 10% of the training split as validation set and keep the rest as training set.

Early Stopping. We use the validation set to estimate the final test accuracy and perform early stopping during the network training to avoid over-fitting. In particular, we stop training the classification network if the validation performance stops improving after a specific number of rounds (specified as a hyperparameter).

A.2 HYPERPARAMETERS FOR EACH EXPERIMENT SETTING

We conduct our experiments in the following four settings: C-I, C-II, A-I, and B-I.

We provide the hyperparameters shared across these settings in Table 3, and discuss setting specific hyperparameters as follows. For setting A-I, and B-I, we use cosine annealing learning rate scheduler with $T_{max} = 200$. For setting C-I and C-II, we start from the learning rate provided in Table 3, and decrease it by a factor of 0.1 every 20 epochs.

A.3 ADDITIONAL DETAILS ON BASELINES

In this section we discuss additional implementation details of Partitioned Coreset/BADGE sampler and VAAL sampler.

Partitioned Coreset/BADGE Sampler. To enable Coreset and BADGE sampler to run on ImageNet, we modify each algorithm to allow to scale, inspired by the approach in Citovsky et al. (2021). At each AL round we partition each of D_U^k and D_L^k into 10 random partitions. We then take one partition from each and combine them into 10 partitions, say $P_1, \ldots, P_{10}$, then run coresets or BADGE separately on each partition using $b/10$ budget.

For BADGE we use global average pooling on the gradient embeddings to reduce their dimension to 512.

VAAL Sampler. We follow the repository provided by the original paper Sinha et al. (2019) to implement VAAL sampler. Since the architecture of the Variational Auto-Encoder (VAE) provided in the repository fails to handle ImageNet naturally, we instead use the VAE architecture in Tolstikhin et al. (2017) and follow the paper to calculate the unsupervised loss with randomly-cropped 64×64 patches instead of the full original images. Also, we perform a single VAE optimizer step for every classifier optimizer step. The original Sinha et al. (2019) paper does not comment on this, however, their codebase performs two VAE optimizer steps for every classifier optimizer step.

B DISTRIBUTION ANALYSIS WITH ENTROPY

Figure 6 displays the entropy of the class distributions at every AL round. Higher entropy is desirable as it indicates more balanced sampling.

Settings	$|s^0|$	b	Epochs	ESP	Batch size	Optimizer	Learning rate	Weight Decay	Momentum
A-I	1000	1000	200	50	128	SGD	$1e^{-3}$	$5e^{-4}$	0.9
B-I	1000	1000	200	50	128	SGD	$2e^{-3}$	0	0.9
C-I	30000	10000	60	30	128	SGD	$1e^{-3}$	0	0.9
C-II	30000	10000	60	30	128	SGD	15	$1e^{-4}$	0.9

Table 3. Hyperparameters used for each setting. $|s^0|$ denotes the initial pool size. b denotes the budget per round, and ESP abbreviates early stop patience.

C   IMAGENET CLASS DISTRIBUTION HISTOGRAMS FOR END-TO-END FINETUNING

Figure 7 contains histograms of the distributions for our end-to-end finetuning ImageNet experiments.

C.1   ADDITIONAL DETAILS ON SELF-SUPERVISED PRETRAINING

We use Moco-V2 Chen et al. (2020b) for all ImageNet experiments, i.e. setting C-I and C-II. We use SimCLR Chen et al. (2020a) for all CIFAR-10 experiments, i.e. setting A-I, and B-I.

As described in the main paper, we train the backbone of the network using respective self-supervised learning method on all available unlabeled data, including the random initial pool but not the corresponding label. For ImageNet experiments, the backbone is trained with Moco-V2 for 800 epochs using SGD. The SGD optimizer uses batch size of 256, learning rate of 0.03, weight decay of 0.0004 and momentum of 0.9. For further detail please refer to the original implementation. For CIFAR-10 experiments, we train the backbone with SimCLR for 1000 epochs with Adam opimizer Cao et al. (2019a). The Adam optimizer uses batch size 256, learning rate of 0.0003, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We also use cosine learning rate schedule.

D   LIMITATIONS

Plotting the test accuracy as a function of exhausted budget is common practice in AL research. However, these experiments are difficult to produce correctly as they are computationally expensive and very sensitive to hyperparameters Anonymous (2022); Beck et al. (2021). We conduct thorough experiments over an extensive set of hyperparameters to ensure fair comparisons. We summarize our findings below.

1. **Training hyperparameters at every round.** This includes the choice of optimizer, learning rate, regularization, and early stopping hyperparameters. It is necessary to train the network to saturation at every round with varying amounts of training data; otherwise, a fair comparison of AL algorithms would not be possible. In fact, if the network is not trained to saturation at a round k, the AL algorithm will not query an optimal set s^k, which will in turn affect the distribution of D_L^{k+i} for all subsequent rounds $k+i$, $i > 1$ Anonymous (2022).

2. **Initial Budget.** s^0 is randomly selected, therefore if the dataset is class balanced, s^0 will be relatively balanced. If s^0 is large, it will take many rounds of querying before we can notice performance differences between AL algorithms that select balanced data and those that don't. It is therefore important to monitor the distribution of D_L^k along with the accuracy of the model at each round before drawing conclusions about performance.
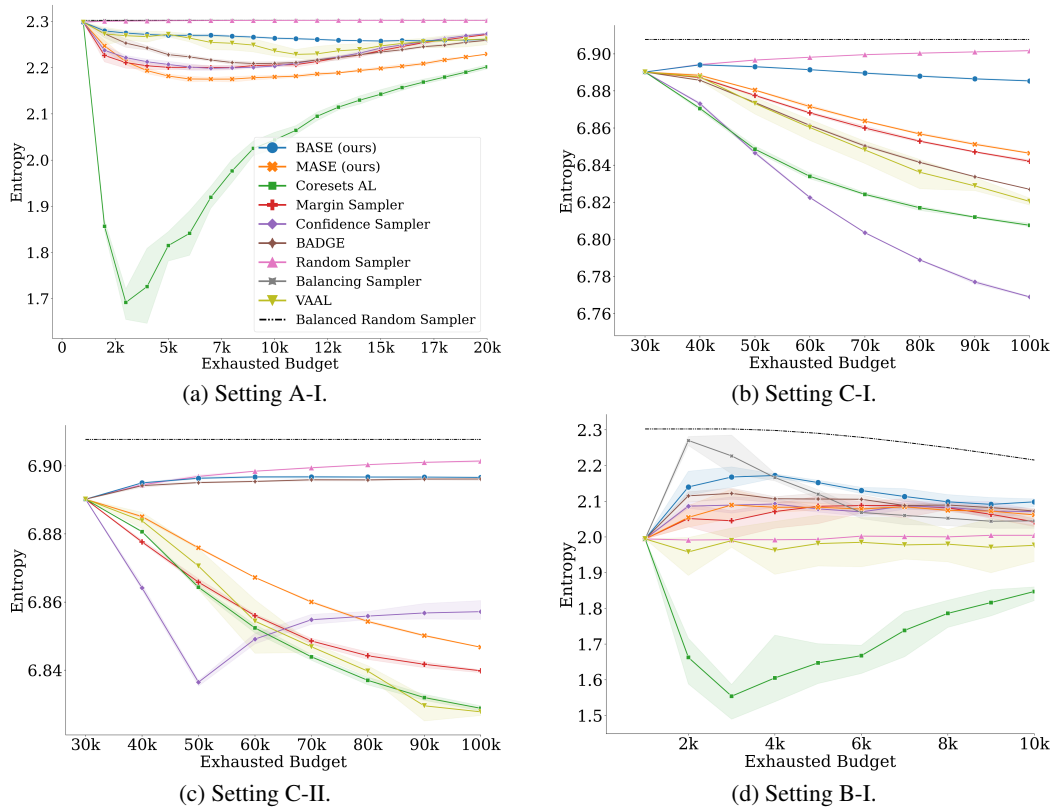
E   TIME COMPLEXITIES

(a) Setting A-I.

(b) Setting C-I.

(c) Setting C-II.

(d) Setting B-I.

Figure 6. Class distribution entropy of D_L^k at different active learning rounds k.

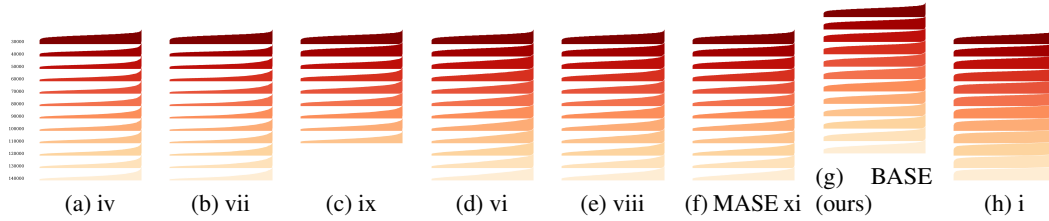(a) iv (b) vii (c) ix (d) vi (e) viii (f) MASE xi (g) BASE (ours) (h) i

Figure 7. Setting C-I. The distribution of D_U^k at every AL round for different strategies on ImageNet in the end-to-end finetuning setting. All experiments start with the same randomly selected subset s^0. The x-axis is sorted for each histogram (every row in every subplot) from least queried class to most queried class. The height of the histogram at a given location on the x-axis indicates the proportion of the examples sampled from that class. BASE is visibly the most balanced strategy after random sampling.

Table 4. Time complexities. d' is the feature space dimension. On ImageNet using ResNet-50, $|D_U^k| \approx 10^6$, features $d' = 2048$, gradient embeddings $d'' = 2048 \times 10^3$, and $C = 10^3$. In our experiments, $b = 10^3$.

Algorithm	Time Complexity
Margin Sampler	$\mathcal{O}(C \cdot \log(b) \cdot |D_U^k|)$
Confidence Sampler	$\mathcal{O}(C \cdot \log(b) \cdot |D_U^k|)$
Approx. Coreset	$\mathcal{O}((b + |D_L^k|) \cdot d' \cdot |D_U^k|)$
BADGE	$\mathcal{O}(C \cdot (b + |D_L^k|) \cdot d'' \cdot |D_U^k|)$
Balancing Sampler	$\mathcal{O}(C \cdot b \cdot d' \cdot |D_U^k|)$
BASE (ours)	$\mathcal{O}(C \cdot (d' + \log(b)) \cdot |D_U^k|)$