On Initial Pools for Deep Active Learning

Akshay L Chandra* AKSHAYCHANDRA@IITH.AC.IN
Sai Vikas Desai* CS17MTECH11011@IITH.AC.IN
Chaitanya Devaguptapu* CS19MTECH110025@IITH.AC.IN
Vineeth N Balasubramanian VINEETHNB@CSE.IITH.AC.IN
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India

Abstract

Active Learning (AL) techniques aim to minimize the training data required to train a model for a given task. Pool-based AL techniques start with a small initial labeled pool and then iteratively pick batches of the most informative samples for labeling. Generally, the initial pool is sampled randomly and labeled to seed the AL iterations. While recent studies have focused on evaluating the robustness of various query functions in AL, little to no attention has been given to the design of the initial labeled pool for deep active learning. Given the recent successes of learning representations in self-supervised/unsupervised ways, we study if an *intelligently sampled* initial labeled pool can improve deep AL performance. We investigate the effect of intelligently sampled initial labeled pools, including the use of self-supervised and unsupervised strategies, on deep AL methods. The setup, hypotheses, methodology, and implementation details were evaluated by peer review before experiments were conducted. Experimental results could not conclusively prove that intelligently sampled initial pools are better for AL than random initial pools in the long run, although a Variational Autoencoder-based initial pool sampling strategy showed interesting trends that merit deeper investigation.

Keywords: Pre-registration, Machine Learning, Active Learning, Initial Pools

1. Introduction

With the success of convolutional neural networks (CNNs) in supervised learning on a wide range of tasks, several large and high-quality datasets have been developed. However, data annotation remains a key bottleneck for deep learning practitioners. Depending on the task, data annotation cost may vary from a few seconds to a few hours per sample and, in many real-world scenarios, supervision of domain experts is necessary (Bearman et al. (2016)). Active learning (AL) methods aim to alleviate the data annotation bottleneck by labeling only a subset of the most informative samples from a large pool of unlabeled data. Various query methods (Gal et al. (2017); Sener and Savarese (2018); Beluch et al. (2018); Yoo and Kweon (2019); Sinha et al. (2019); Mottaghi and Yeung (2019)) have been recently proposed for AL in the context of deep neural network (DNN) models (deep AL). The problem is

* Equal Contribution.

important in deep AL, since DNN models require large amounts of labeled data to learn. In this work, we focus on the popular pool-based AL framework, which starts with a small initial labeled pool after which AL is performed in multiple *sample-label-train* cycles.

AL has been well-explored in the context of traditional (shallow) machine learning methods (Settles (2009)). Generally, before starting the AL cycles, a small randomly chosen subset of a dataset (with size around 1-10% of the entire dataset), typically called the *initial pool*, is labeled first to train an initial model. Across all AL efforts so far, to the best of our knowledge, the initial pool is always sampled randomly and labeled (Gal et al. (2017); Sener and Savarese (2018); Beluch et al. (2018); Yoo and Kweon (2019); Sinha et al. (2019); Lewis and Gale (1994)). This initial pool design strategy has generally worked well for AL in traditional/shallow ML models. However, the success of AL in DNNs has not been convincing yet, especially when such models are trained on large-scale datasets. On one hand, while there have been several encouraging newly proposed deep AL methods, deeper analysis of those methods in (Munjal et al. (2020); Mittal et al. (2019); Lowell et al. (2019)) show that AL struggles to outperform random sampling baselines when slight changes are made to either datasets (class-imbalance) or training procedures (data augmentation, regularization, etc.). Interestingly, to the best of our knowledge, the design of better initial labeled pools received no attention by the deep AL community. Considering the tremendous success of self-supervised learning methods in recent years (Chen et al. (2020); Gidaris et al. (2018); Caron et al. (2018); Pathak et al. (2016); Noroozi et al. (2017); Noroozi and Favaro (2016); Doersch et al. (2015)), we ask the question if choosing an initial labeled pool intelligently can improve AL performance.

In our work, we propose to perform an empirical study of deep AL methods while using initial labeled pools, sampled using methods other than random sampling. To investigate the effect of *intelligently* sampled initial labeled pools on deep AL methods, we propose two sampling techniques, leveraging state-of-the-art self-supervised learning methods and well-known clustering methods. In particular, we propose the following ways of choosing the initial pool:

- Sample datapoints that a state-of-the-art self-supervised model finds *challenging*, as observed using the trained model's loss on the data.
- Cluster the unlabeled pool first and then perform sampling across each cluster. Equal proportions of datapoints are sampled from each cluster to make sure the chosen samples span the entire dataset.

We hypothesize that AL methods (we focus our efforts on deep AL methods) can benefit from more intelligently chosen initial pools, thus eventually reducing annotation cost in creation of datasets. Our empirical study will seek to address the following specific questions:

- Can pool-based deep AL methods leverage design of intelligently sampled initial pools to improve AL performance?
- Can we exploit latest advancements in self-supervised learning to boost deep AL performance with no additional labeling cost?
- Are some initial pools better than others? What makes an initial pool *good*?

In a realistic training setting with measures to avoid overfitting (*i.e.* regularization, batch norm, early stopping), we hypothesize that the generalization error of AL models starting with our initial pools will be lower than those of AL models starting with random initial pools, across AL cycles. However, as AL cycles increase, we expect to see shorter margins of error difference as the effect of our initial pools on the model performance could diminish with increase in labeled pool size. Studying the use of unsupervised/self-supervised learning in later epochs could be an interesting direction of future work. If initial pools do contribute to better model performances, our work could make a positive contribution to: (i) boosting AL performance with no additional annotation; (ii) developing datasets with lesser annotation cost in general; and (iii) promoting further research in the use of unsupervised learning methods for AL. On the other hand, if random initial pools perform better than our initial pools, the community will still have useful insights about this rather unexplored part of AL through this study.

2. Related Work

Analysis of Active Learning: In recent years, previous works have evaluated the robustness and effectiveness of deep AL methods for various tasks. Lowell et al. (2019) first reported some obstacles of deploying AL in practice by empirically evaluating consistency of AL gains over random sampling and transferability of active samples across models. Along the same lines, Mittal et al. (2019) evaluated the performance of deep AL methods under data augmentation, low-budget regime and a label-intensive task of semantic segmentation. More recently, Munjal et al. (2020) comprehensively tested the performance variance of deep AL methods across 25 runs of experiments. They considered various settings such as regularization, noisy oracles, varying annotation and validation set size, heavy data augmentation and class imbalance. However, none of these efforts have considered varying the sampling strategy for the initial pool.

Exploiting Unlabeled Data: Our focus is on finding out if initial pools with certain *desirable* qualities can bolster AL performance. We exploit self-supervised pretext tasks to sample the initial pool more intelligently. Previous works have successfully managed to integrate unlabeled data into AL using self-supervised learning and semi-supervised learning. Siméoni et al. (2019) showed that initializing the target model with the features obtained from self-supervised pretraining gives AL a kick-start in performance. Contemporaneously, Mottaghi and Yeung (2019) also used this technique in combination with a GAN based AL method and reported SOTA results on SVHN, CIFAR-10, ImageNet, CelebA datasets. This is enough evidence that exploiting self-supervised learning methods can boost AL performance, but the cited works operate in the model weight space. The importance of good initialization in weight space (Mishkin and Matas (2016); He et al. (2015); Glorot and Bengio (2010)) is well understood by the deep learning community; To the best of our knowledge, there have been no efforts in understanding the importance of good initialization in data space for deep AL methods. In case of traditional AL (before deep learning's popularity), there have been handful of encouraging works that support our hypothesis (Kang et al. (2004); Hu et al. (2010)). Both these works use k-means clustering to initialize the initial label pool,

k-nearest neighbor algorithm for training and report better AL performance on small scale text classification tasks.

Model Loss for AL: In our work, we use a trained model's loss to identify the most *informative* unlabeled samples. Existing AL methods largely rely on using the target model's loss for active sampling. Settles et al. (2007) first proposed an AL framework by calculating Expected Gradient Length (EGL) where the learner queries an unlabeled instance which, if labeled and added to the labeled pool, would result in the new training gradient of the largest magnitude. More recently, Yoo and Kweon (2019) proposed a loss prediction module which is attached to the target network to predict the loss value of unlabeled samples. In contrast to these methods, we strictly rely on a self-supervised model's loss, instead of the target model, since the initial pool needs to be selected/sampled, before any model is trained on the target data.

3. Methods and Experimental Protocol

In this section, we describe: (i) the notations and setting for pool-based AL cycles; (ii) our strategies for sampling the initial pool; and (iii) the AL methods that are subsequently used to build on top of the initial labeled pool. Implementation details and other considered additional experiments and ablation studies are mentioned at the end of this section.

3.1. Pool-based Active Learning Setting

Given a dataset $\mathcal{D}$, we split it into train (T_r), validation (V), and test (T_s) sets. At the beginning, the train set is also treated as an unlabeled (U) set, from which samples are moved to a labeled set (L) after every AL cycle. Pool-based AL cycles operate on a set of labeled data $L_0{=}\{(x_i, y_i)\}_{i=1}^{N_L}$ and a large pool of unlabeled data $U_0{=}\{x_i\}_{i=1}^{N_U}$, and model Φ_0 is trained on L_0 in every AL cycle. In our setting, given $L_0 = \emptyset$ to start with, a sampling function $\Psi(L_0, U_0, \Phi_0)$ parses $x_i \in U_0$, and selects k (budget size) **samples**. These samples are then labeled by an **oracle** and added to L_0, resulting in a new, extended L_1 labeled set, which is then used to **retrain** Φ. This cycle of **sample-label-train** repeats until the sampling budget is exhausted or a satisfying performance metric is achieved. In our case, we populate L_0 using our proposed methods, discussed in Section 3.2. Sec 3.3 describes the query methods we use to perform the traditional AL cycles after this initial pool selection. We can confirm that there exists a good initial pool if the generalization error of models starting with our initial pools is lower than those of models starting with random initial pools across the AL cycles.

3.2. Proposed Initial Pool Sampling Strategies

We now describe our initial pool sampling strategies, which were briefly stated in Section 1.

Our methods are fundamentally motivated by the hypothesis that samples considered *challenging* for an unsupervised/self-supervised setting can help bootstrap AL methods through a more intelligent, well-guided choice of an initial labeled pool.

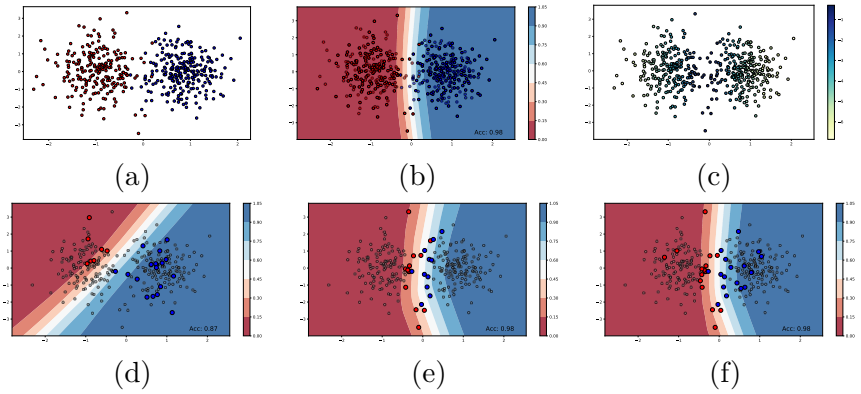

Figure 1: Illustration of query strategies in traditional pool-based AL: (a) A toy dataset of 500 instances, evenly sampled from two class Gaussians; (b) Decision boundary of a logistic regression model trained on the dataset; (c) Trained model's *CrossEntropy* loss on training instances; Decision boundary of logistic regression models trained on 35 instances chosen (d) *randomly* (e) using Least Confidence method Lewis and Gale (1994) (f) using Max-Entropy method Shannon and Weaver (1963); Best viewed in color. Labeled instances are emphasized for clarity.

3.2.1. Self-Supervision Methods

As shown in Figure 1, well-known pool-based AL methods rely on choosing samples with a high uncertainty of the target model trained on an initial labeled pool.

Since our focus is on choosing an initial labeled pool where there is no target model, we cannot use these AL query methods since we do not have access to labels to calculate the supervised model's loss on training data. We hence train a self-supervised model on the entire unlabeled pool and identify samples as "challenging", where the self-supervised model's loss is relatively higher than that of others. The recent success of self-supervised learning in learning useful data representations (Doersch et al. (2015); Noroozi and Favaro (2016); Noroozi et al. (2017); Gidaris et al. (2018); Caron et al. (2018); Chen et al. (2020)) motivates us to hypothesize that such a model could help us sample the most *informative* datapoints without any supervision.

Let τ be any self-supervised task with an objective to minimize the loss function $\mathcal{L}$. Let θ be trained weights obtained by solving τ on the unlabeled data pool U. We want the oracle to label and populate the initial pool L_0 with datapoints sampled by solving:

$$\arg\max_i \mathcal{L}^\tau(x_i;\theta) \quad \forall x_i \in U \tag{1}$$

Since we are working with a learned model's loss, any self-supervised task can be used, making our proposed method task-agnostic. We have chosen tasks that are simple and easy to interpret, such as image inpainting (Pathak et al. (2016)) and image rotation prediction (Gidaris et al. (2018)). For example, in case of the rotation prediction task, our strategy can be summarized as: *if a trained rotation predictor struggles to rightly predict the rotation of a sample, even after looking at it while training, then it is a hard sample - thus human labeling*

is needed. In addition to the above tasks, we will also train a Variational Autoencoder (VAE) (Kingma and Welling (2014)) as one of our tasks where datapoints with highest loss *i.e.* hard to reconstruct images are sampled for the initial pool. We want to do this to understand how complexity of self supervised tasks (*e.g.* image inpainting task is more complex than VAEs) relates to efficiency of the sampled initial pool using those tasks.

3.2.2. Unsupervised Learning (Clustering) Methods

Sampling bias is the most fundamental challenge posed by AL especially in case of uncertainty based AL methods (Dasgupta (2011)). Assume AL is performed on a dataset with data distribution $\mathcal{D}$. But as AL cycles proceed, and datapoints are sampled and labeled based on increasingly confident assessments of their informativeness, the labeled set starts to look less like $\mathcal{D}$. This problem is further exacerbated by highly imbalanced real-world datasets where random initial samples, with high probability, may not span the entire data distribution $\mathcal{D}$. To overcome this, several works proposed diversity based methods (Sener and Savarese (2018); Sinha et al. (2019)) whose fundamental goal is to sample unlabeled datapoints from a non-sampled area of $\mathcal{D}$ such that all areas of $\mathcal{D}$ are seen by the target model. Motivated by these methods and their success, we propose a clustering-based sampling method for choosing the initial pool such that the sampled points spans all area of $\mathcal{D}$ (*i.e.*, all clusters) even before AL starts. In a way, this is analogous to exploration in AL (Bondu et al. (2010)), albeit in an unsupervised way.

We assume that number of classes to be labeled (K) in the dataset $\mathcal{D}$ is known apriori. If a clustering algorithm is applied on the unlabeled data $U=\{x_i\}_{i=1}^{N_U}$ to obtain K clusters and every datapoint x_i is assigned only one cluster C_j, we get K disjoint sample sets $C = \{C_1, C_2, ..., C_K\}$. If the initial pool budget is B samples, we sample $\frac{B}{K}$ datapoints from each cluster. Equal weight is given to each cluster to make sure initial pool is populated with datapoints that span the entire $\mathcal{D}$. As another variant to this method, we will also experiment by sampling $\frac{|C_j|*B}{N_U}$ datapoints from each cluster, keeping the original cluster proportions intact. We will use DeepCluster (Caron et al. (2018)) and k-means as clustering methods in our experiments.

3.3. Active Learning Query Methods

In order to study the usefulness of the choice of the initial labeled pool across AL methods, we need to study different AL query methods in later cycles of model updation. Modern pool-based AL methods may be broadly classified into three categories. We will evaluate the effectiveness of our sampling methods on AL methods from all three categories:

- **Uncertainty Sampling:** Least Confidence (LC) (Lewis and Gale (1994)), Max-Entropy (ME) (Shannon and Weaver (1963)), Min-Margin (MM) (Scheffer et al. (2001)) & Deep Bayesian AL (DBAL) (Gal et al. (2017))
- **Diversity Sampling:** Coreset (greedy) (Sener and Savarese (2018)) & Variational Adversarial AL (VAAL) (Sinha et al. (2019))
- **Query-by-Committee Sampling:** Ensemble with Variation Ratio (ENS-varR) (Beluch et al. (2018)) (3 ResNet18 models) & ensemble variants of Least Confidence (ENS-LC), Max-Entropy (ENS-ME) and Margin Sampling (ENS-MM)

All the above methods are already implemented in the AL toolkit offered by Munjal et al. (2020), and we will leverage it to study the methods.

4. Implementation Details

Following recent deep AL efforts, we will use MNIST, CIFAR-10, CIFAR-100 and Tiny ImageNet-200 (Le and Yang (2015)) datasets in our experiments. We use the AL methods, model architectures, data augmentation schemes and implementation details from Munjal et al. (2020) for our experiments.

For all datasets, we plan to tune hyperparameters using grid search. However, going by previous works, we expect to use an Adam optimizer (Kingma and Ba (2015)) across the datasets. For datasets CIFAR-10 and CIFAR-100, we expect to use learning rate (lr), weight decay (wd) from Munjal et al. (2020) - ($lr = 5e^{-4}$, $wd = 5e^{-4}$) and ($lr = 5e^{-4}$ and $wd = 0$) respectively. For all datasets, we augment the data with random horizontal flips ($p = 0.5$) and normalize them using statistics provided in [1], [2]. We will use ResNet18 (He et al. (2016)) for all our experiments.

AL Details: As usually done in most AL work, we will initialize L_0 with 10% of the unlabeled set U and in every AL cycle 10% of the original unlabeled set U will be sampled, labeled and moved to labeled set L_i. However, we expect some changes in AL cycle details due to irregularities between datasets (*e.g.* MNIST is easier to learn compared to Tiny ImageNet) and those changes will be reported appropriately post-experiments.

Performance Metrics: We will measure accuracy on the test set after every AL cycle (including after the choice of the initial labeled pool). Our initial pool sampling strategies will be compared against a random selection of the initial pool (the default option used today), and all our results will be reported (as mean and std) over 5 trials to avoid any randomness bias in the results.

We also plan to visualize the chosen initial labeled pool using t-SNE embeddings in case this provides any understanding of sampling strategies that work best. We will also examine overlap between every labeled pool acquired during all AL cycles when our initial pool sampling strategy is used against a random choice. This would allow us to know if initial pool played any role in altering the labeled pools (for better or worse).

4.1. Additional Experiments

In practice, populating the initial pool only with *challenging* datapoints may not be fully conducive for learning. Hence, we plan to follow Roy et al. (2018) and split the sorted list obtained by solving Eqn (1) into n equal-sized bins. If the initial pool budget size is B, we query $\frac{B}{n}$ highest scored images from the top $(n-1)$ bins (hard samples) and $\frac{B}{n}$ lowest scored images from the last bin (easy samples). So the resultant batch contains images from different regions of the score space. In the experiments, we will use 2, 5 and 10 as the values of n.

1. https://github.com/pytorch/examples/
2. https://github.com/kuangliu/pytorch-cifar

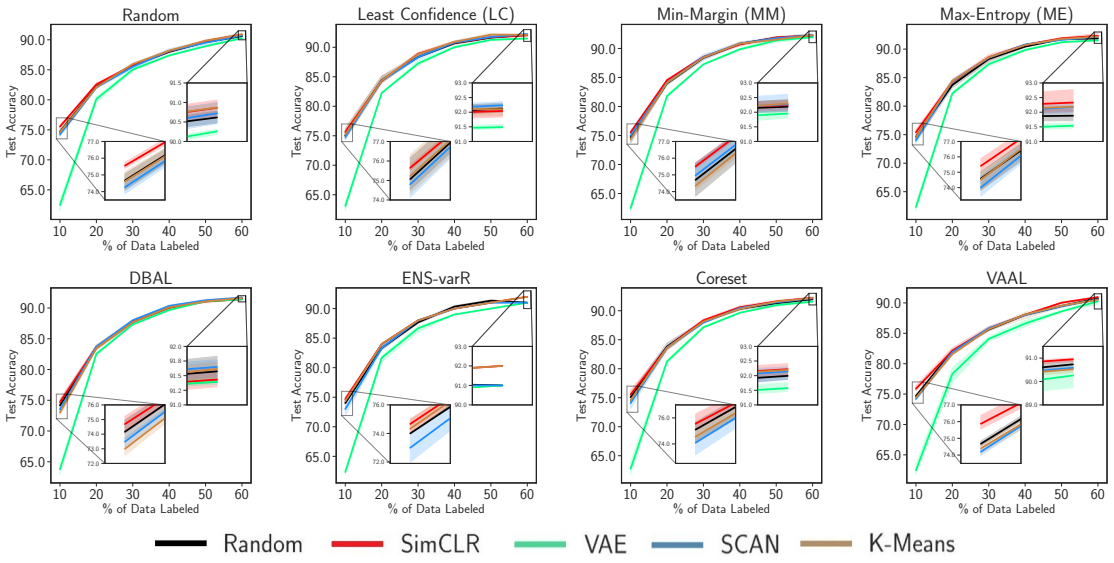

Figure 2: Performance of each active learning query method with different initial pool sampling strategies on CIFAR-10. There are 8 graphs shown, one for each active querying method as mentioned in the graph titles. Each colored line in the graph corresponds to an initial pool sampling method, as shown in the legend.

Additionally, we will test the usefulness of our sampling methods on AL for imbalanced data. For this, we will follow Cui et al. (2019) to simulate a long-tailed distribution of classes on CIFAR-10, by following power law.

5. Experimental Results

In this section, we first document the modifications to the original experimental protocol. Then, we present the experimental results on MNIST, CIFAR-10, CIFAR-100 and Tiny ImageNet datasets. Then, we discuss our experimental findings and evaluate the extent to which intelligently sampled initial labeled pools help boost AL performance[3]. Finally, we provide more training details needed for reproducing our results.

5.1. Modifications to the Original Proposal

Switching to Better Performing Tasks: We initially proposed to use image rotation prediction (Gidaris et al. (2018)) and image inpainting (Pathak et al. (2016)) as the self-supervision tasks. To make the experimental setting strong, we switched to a more popular and stronger recent self-supervision method - SimCLR (Chen et al. (2020)). In case of unsupervised clustering-based methods, we switched from DeepCluster (Caron et al. (2018)) to a more recent and state-of-the-art clustering method - SCAN (Van Gansbeke et al. (2020)), which builds on top of SimCLR as a pretext task.

3. Our code is available at https://github.com/acl21/init-pools-dal

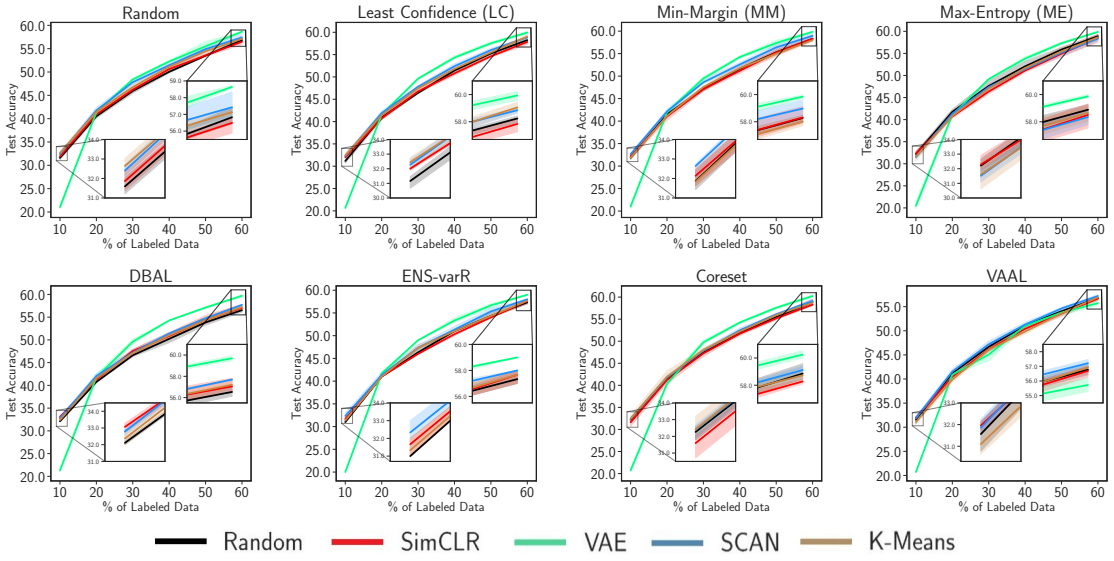

Figure 3: AL Performance of each active learning query method with different initial pool sampling strategy on CIFAR-100.

Adding Passive Learning to Experiments List: In addition to the AL query methods, we run passive learning (random sampling) cycles starting with intelligently sampled initial pools. This passive learning configuration is equivalent to placebo control group where the intervention wouldn't have direct impact on the final outcome, since we sample data randomly each episode. This is done to check if the performance gains on active learning due to an intelligently labeled initial pool are coincidental.

Miscellaneous Changes: Due to time and computational constraints, we repeated each experiment 3 times instead of the proposed 5 repetitions. We report results on seven out of ten proposed AL query methods, excluding ensemble variants of Least Confidence (ENS-LC), Max-Entropy (ENS-ME) and Min-Margin (ENS-MM). For the same reason, we could not perform the additional binning experiments described in Section 4.1. We also do not report results of VAAL query method on Tiny ImageNet since one AL cycle with 5 episodes (1 run of experiment) executed for over 100 hours on a single GeForce GTX 1080 Ti GPU. On MNIST, we run 10 episodes of AL starting with 60 instances (0.1%) in the initial pool and set the AL budget to 60 as well.

Using grid search, we obtained hyperparameters which are better than the ones mentioned in the proposal. We report the final hyperparameter choices in section 6.

5.2. Initial Pool Sampling Details

For completeness, we briefly describe the methods used for our experiments below:

SimCLR: Contrastive learning methods, such as SimCLR (Chen et al. (2020)), learn representations by contrasting positive pairs against negative pairs. Positive pairs include

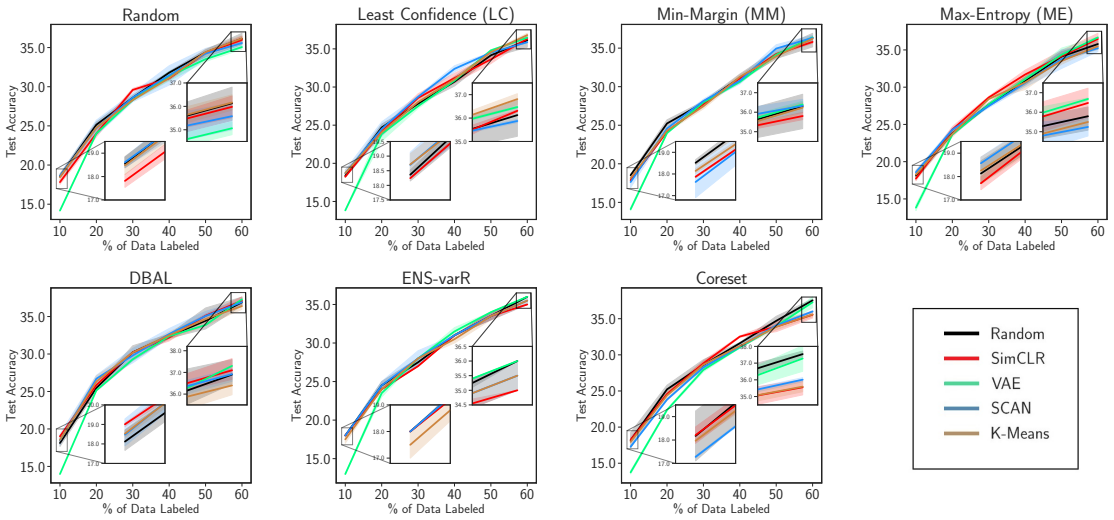

Figure 4: Tiny ImageNet: Our initial pools perform no better than random initial pools across all AL configurations.

input images and their augmented variants. Ideally, a trained SimCLR model should have comparatively low contrastive loss for positive pairs of a given image taken from the unlabeled set. We design our sampling method on this fact. Firstly, we train a ResNet-18 SimCLR model with the recommended augmentations: image horizontal flipping, Gaussian blur, color jitter, and image gray-scale. After training the model, we assign each image in the unlabeled pool a score - model's average contrastive loss between an input image and four of its augmented variants[4]. The higher the average contrastive loss, the *harder* it was for the trained SimCLR model to learn that input, so we sample such images first.

VAE: We train a vanilla VAE model on the entire training data till convergence. We then sample those data points from the training set whose reconstruction error was high post-training. The higher the reconstruction error, the *harder* it was for the model to learn such images, hence we sample them first. We chose VAE in particular to understand how task complexity (the VAE reconstruction task is simpler than SimCLR's) contributes to initial pool efficiency.

SCAN and K-Means: SCAN (Van Gansbeke et al. (2020)) is a state-of-the-art clustering method where feature learning and feature clustering are decoupled. SCAN builds on top of features learned by any self-supervision task (in our case, we used SimCLR). At the end of training, the SCAN model assigns a single cluster to each data point. In case of K-Means, we apply K-Means algorithm to SimCLR-learned feature representations to get cluster assignments. Once again, we chose these two clustering methods (one very simple, K-Means, and one more sophisticated, SCAN) to understand the role of model complexity w.r.t. the effectiveness of the initial pool.

4. MNIST dataset has gray-scale images so we average the contrastive loss over the other three augmentations.

5.3. Results

5.3.1. Main Experiments

Figures 2-5 depict the main results of our experimental study on finding good initial pools for AL. We first describe how the results are illustrated in Figures 2-5. We have one figure for each of the following four datasets: CIFAR-10, CIFAR-100, Tiny ImageNet and MNIST. Each plot inside the figures depicts the performance of one AL method with various initial pool techniques. For instance, the plot titled VAAL (second row, right most) in Figure 2 shows the performance of models trained on data sampled by VAAL's query method in each episode, however initiated with different initial pool strategies (indicated by different colored lines). For example, the red lines show that AL methods were started using the SimCLR-based initial pool strategy. The plots are conventional AL plots where the x-axis represents the percentage of labeled data used to train the model, and y-axis represents the model's performance on the test set.

We now briefly analyze the results of each initial pool sampling strategy.

SimCLR: In Figure 2 (CIFAR-10), before the first episode, models trained with SimCLR-sampled initial pools show better performance than models trained on a randomly generated initial pool across all eight configurations, including passive learning. However, this performance gain in the beginning of the AL cycles did not contribute to the model in picking better active samples. We can see that models starting with SimCLR-based initial pools performed similar to the models which started with random initial pools at each episode of the AL cycles. Similarly, on CIFAR-100, we see in Figure 3 that the models starting with SimCLR sampled initial pools perform either same or worse than the models starting on random initial pools at both ends of the AL cycles across all eight configurations. On Tiny ImageNet (Figure 4) and MNIST (Figure 5), we see the same trend as that of CIFAR-100's.

SCAN and K-Means: Across all datasets, none of the two clustering methods: SCAN and K-Means, show signs of contributing to better model performances compared to random initial pools.

VAE: Perhaps the most surprising behaviour we noticed among all the methods was how VAE-sampled initial pools worked. Models trained with VAE sampled initial pools consistently underperformed in the first episode of the AL cycles across all four datasets. On CIFAR-10, in the first episode, the average test accuracy difference between models trained with VAE initial pools and other initial pools was 12%. Similarly, there was a 11% difference in case of CIFAR-100, 4.5% in case of Tiny Imagenet and 18% in case of MNIST. We suspect this is due to the difference in models used for initial pool sampling and active learning. It has been empirically shown that data points actively sampled by one model, say VGG16, do not transfer well to another model, say ResNet-18 (Lowell et al. (2019); Munjal et al. (2020)). To allow for smooth transfer of samples, we used the same ResNet18 model for training SimCLR, SCAN and active learning episodes. However, following general trends of use of VAEs, we use a simple VAE model with 4 convolutional layers each in the encoder and the decoder, which may have resulted in this significant difference.

At the end of all AL cycles, on CIFAR-10, Tiny Imagenet and MNIST, we see that all initial pools converge to largely similar test accuracy, suggesting no significant improvement

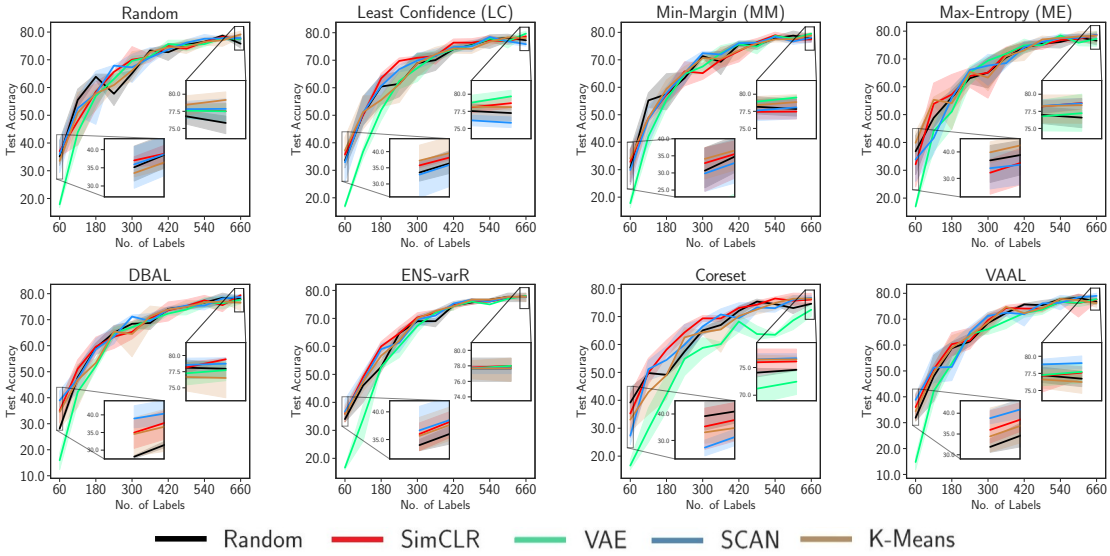

Figure 5: MNIST.

in AL performance. But we see a different outcome in the case of CIFAR-100 (Figure 3). On CIFAR-100, models starting with VAE-sampled initial pools, despite the bad start, ultimately outperform the models starting with the other four initial pools in six out of the seven configurations. In the passive learning configuration (control experiment), we notice that VAE appears to be outperforming others but that happens only in the final episode of the AL cycle, suggesting that this performance gain in six configurations was not a mere coincidence. The models starting with VAE start to outperform their random counterparts right after the second episode (20%), and we notice the VAE curve starting to diverge from others. However, this behavior was only seen on CIFAR-100.

To summarize the findings, our proposed methods could not conclusively prove the existence of *good* initial pools that help AL methods in the long run, although the use of VAE-based initial pool strategy showed some interesting trends.

What explains the odd behavior of VAE sampled initial pools on CIFAR-100? To investigate the reason behind the odd behavior of models started with VAE-based initial pools, we study the class distribution of initial pools obtained by 4 methods - VAE, SCAN, SimCLR and K-Means. To this end, we picked initial pools from the DBAL experiment.

Looking at the class frequencies of all CIFAR-100 initial pools in Figure 6, we notice a clear difference between the VAE-sampled initial pool and the others. VAE-sampled initial pool has more class imbalance and is particularly emphasizing on images from specific classes. To verify if the VAE-based initial pool sampling technique is biased towards difficult classes, we created two sets of CIFAR-100 classes: (1) Top 10 classes sampled by VAE, (2) 10 classes with least per-class test accuracy w.r.t. the model in the final AL episode. We use (2) as a proxy for "difficult" classes. We observed that both these sets have 4 classes in common. While this overlap is not high enough to conclude that VAE sampling is biased towards difficult classes, it nevertheless is an interesting future direction to pursue, and merits more study.

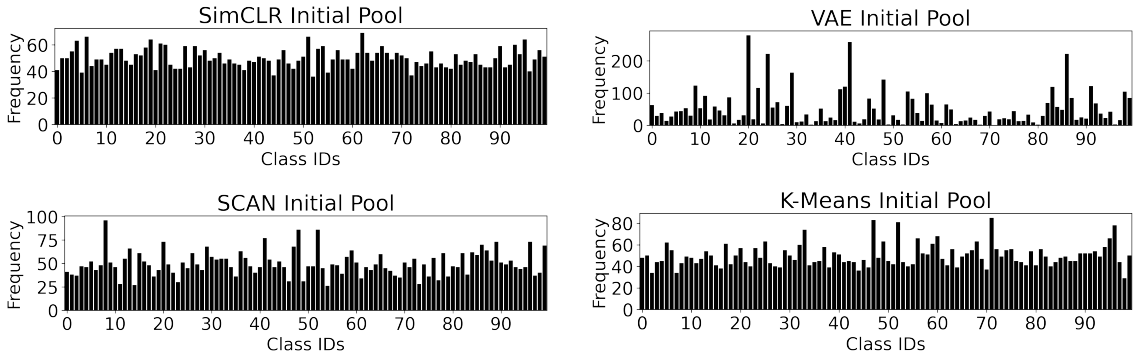

Figure 6: CIFAR-100: Class distribution of initial pools picked by various methods. Note the apparent class imbalance in the initial pool picked by VAE. Is this the reason for the performance gain?

5.3.2. Ablation Experiments

Comparing the Initial Pools: We used SimCLR representations to obtain t-SNE embeddings on all initial pools of a randomly chosen Max-Entropy (ME) experiment on CIFAR-10. The t-SNE plots of 5000 data points are shown in Figure 7. Unsurprisingly, we see no apparent inconsistencies or anomalies in either class distribution or inter-class relationships across the four initial pools except for the VAE-sampled initial pool whose class distribution is noticeably different than the others.

The other four initial pools are nearly identical. A confusion matrix with their overlap statistics, shown in Figure 7, shows that all five initial pools roughly shared approximately 10% of the data points among themselves. Even with nearly 90% of unique data points, all four initial pools contributed to largely similar model generalization error (as seen in Max-Entropy graph of Figure 2).

Low-Budget AL: Is 10% of data points too many for the model? Is that why we are unable to spot any potential performance differences between the four mostly unique initial pools? To find out if a low query budget can help spot performance differences, we repeated our experiments on CIFAR-10 for Max-Entropy (ME), Least Confidence (LC) and Deep Bayesian (DBAL) AL query methods but with just 1000 samples (2% of the overall dataset size) in the initial pool. We set the AL budget to 1000 and allowed the AL cycles to run up to 10 episodes (22% of the overall dataset size). The results of these experiments (averaged over 2 runs) are shown in Figure 8. All three AL methods benefit from VAE-sampled initial pools, albeit marginally, while other initial pools do not contribute to any performance gain compared to random initial pools.

Long-Tail CIFAR-10: One of the motivations behind our proposed unsupervised method (Section 3.2.2) was to allow AL cycles to start with a balanced initial pool, which spans the entire data distribution, when dealing with imbalanced datasets. To that end, we created a Long-Tail CIFAR-10 with an imbalance factor of 50 (Cui et al. (2019)). We report the experiment results on three AL methods (ME, LC, DBAL) averaged over 2 runs in Figure

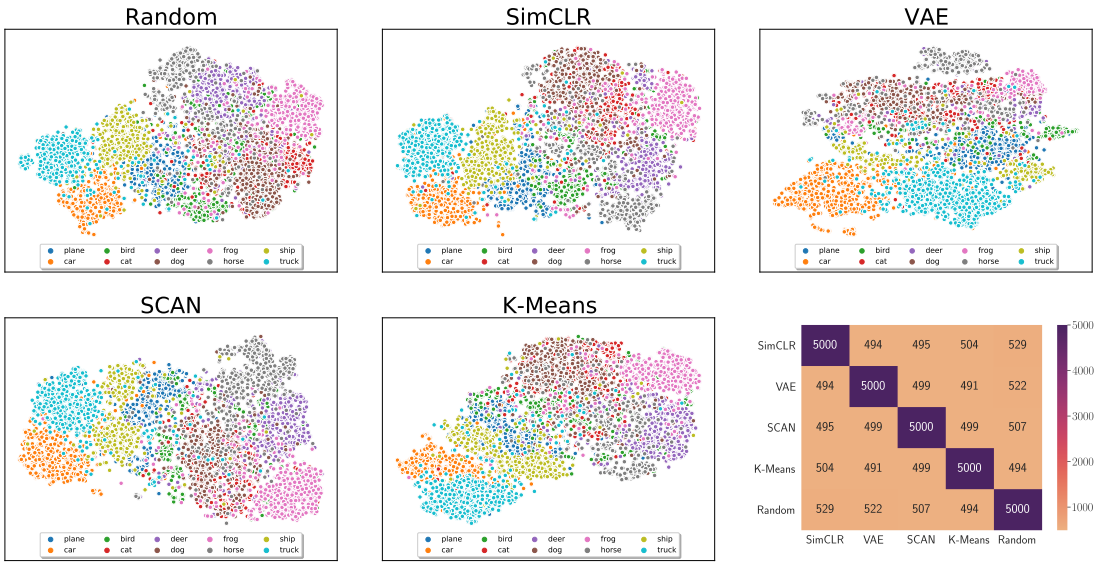

Figure 7: CIFAR-10: Initial pools visualized using t-SNE.

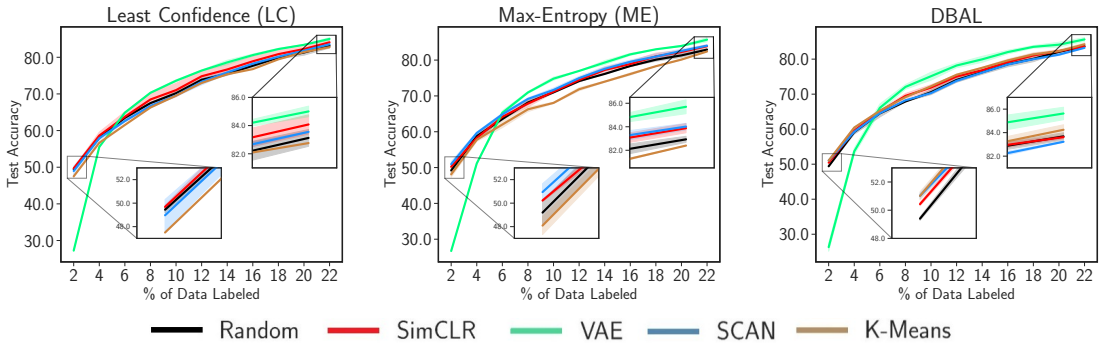

Figure 8: CIFAR-10: In low budget AL setting, only VAE initial pools show marginal performance gains over random initial pools.

9. Surprisingly, our unsupervised initial pool sampling methods did not help the three AL methods. In fact, models trained on SCAN-based initial pools did consistently worse than models trained on random initial pools. Once again, VAE-based initial pools positively contribute to three AL methods albeit the performance gain is quite marginal.

6. More Training Details

In this section we mention more training details necessary for reproducing our experiments.

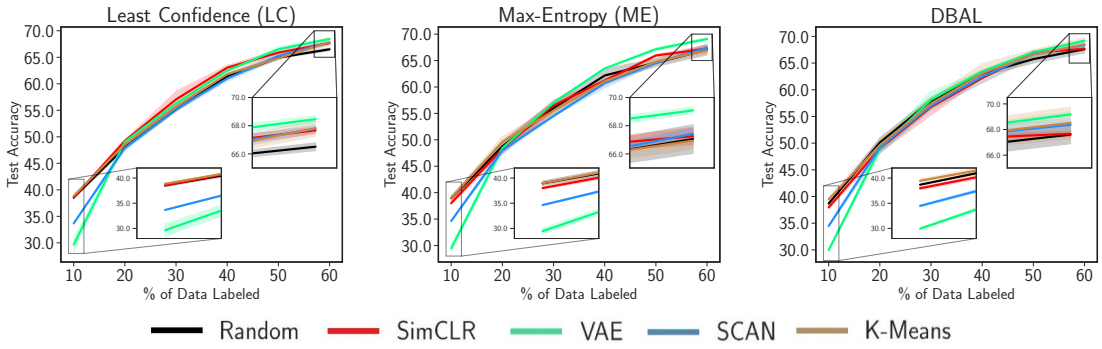

Figure 9: Long-Tail CIFAR-10: Our unsupervised sampling methods (SCAN and K-Means), motivated by this imbalance setting, did not improve LC, ME, DBAL query method performances. In the long run, VAE-based initial pools show marginal performance gains over random initial pools.

6.1. Slightly Modified ResNet18 Model

To add an extra projection layer as the penultimate layer of the model is a convention in self-supervised learning methods (Chen et al. (2020); Grill et al. (2020)). To be consistant with the ResNet18 model used for SimCLR and SCAN training, we added a projection layer to the model just before the final fully connected layer. Projection dimension was set to 128 for MNIST, CIFAR-10/100, 512 for Tiny ImageNet experiments. Also, the official ResNet18 implementation doesn't include any dropout layers in it. To allow for DBAL method to run Monte Carlo simulations, we added a dropout layer with p=0.5 after the flattening layer. We only did this for DBAL experiments.

6.2. Hyperparmeters for AL Training

For all experiments on all datasets, we set momentum = 0.9, $wd = 3e^{-4}$, gamma = 0.1. Other hyperparameter choices are as follows:

Dataset	Epochs	Optimizer	Learning Rate	Scheduler	Batch Size
CIFAR-10/100	200	SGD	0.025	Cosine (0.1)	96
MNIST	100	Adam	0.005	None	64
TinyImageNet	100	Adam	0.001	None	200

Table 1: Hyper-parameters of AL Cycles

6.3. SimCLR, SCAN and VAE Training

For all four datasets we train SimCLR and SCAN with largely similar hyperparameters. We use the official implementation of SCAN[5] (includes SimCLR implementation as well) for training and use their recommended hyperparmeters across all experiments. In case of

5. https://github.com/wvangansbeke/Unsupervised-Classification/

CIFAR-100, we follow the standard practice and group the 100 classes into 20 super classes before training SimCLR (the grouping details can be found in the official SCAN repository). Evaluation metrics of our final SimCLR + SCAN + Self Labeling models are as follows:

Dataset	ACC	NMI	ARI
CIFAR-10	0.70	0.46	0.38
CIFAR-100	0.44	0.41	0.25
MNIST	0.86	0.72	0.72
Tiny ImageNet	0.10	0.07	0.05

Table 2: Final Model Performances after Self-Labeling (SimCLR + SCAN + Self-Label)

In case of VAE training, we trained a Vanilla VAE[6] on the entire training data with hyperparameters as follows: optimizer = Adam, $lr = 0.001$, epochs = 100, momentum = 0.9, $wd = 5e^{-4}$, batch size = 200, for all four datasets. We used 5% of the training data as the validation set. The model weights at epoch with best loss are saved for initial pool sampling.

7. Conclusion

In this paper, we proposed two kinds of strategies – self-supervision based and clustering based – for intelligently sampling initial pools before the use of active learning (AL) methods for deep neural network models. Our motivation was to study if there exist good initial pools that contribute to better model generalization and better deep AL performance in the long run. Our proposed methods and experiments conducted on four image classification datasets couldn't conclusively prove the existence of such good initial pools. However, a surprising outcome of this study was how initial pools sampled with a simple VAE task contributed to improved AL performance, better than more complex SimCLR and SCAN tasks. Even though VAE-based initial pools worked better than random initial pools only on one dataset (CIFAR-100), ablation studies on low budget CIFAR-10 settings as well as on Long-Tail CIFAR-10 point towards potential in VAE-sampled initial pools. Are images that are hard to reconstruct for VAEs good for generalization? Can better generative models like GANs do better than VAEs? We leave this for future work. While our methods and findings could not conclusively prove our hypothesis that AL methods can benefit from more intelligently chosen initial pools, we are optimistic about the potential this research direction holds.

Acknowledgements

We thank DST, Govt of India, for partly supporting this work through the IMPRINT program (IMP/2019/000250). We also thank the members of Lab1055, IIT Hyderabad for engaging and fruitful discussions. Last but not the least, we thank all our anonymous reviewers for their insightful comments and suggestions, which helped improve the quality of this work.

6. https://github.com/AntixK/PyTorch-VAE

References

Amy L. Bearman, Olga Russakovsky, V. Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016.

William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

Alexis Bondu, V. Lemaire, and M. Boullé. Exploration vs. exploitation in active learning : A bayesian approach. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2010.

M. Caron, P. Bojanowski, Armand Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and S. Belongie. Class-balanced loss based on effective number of samples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, 2019.

S. Dasgupta. Two faces of active learning. *Theor. Comput. Sci.*, 412:1767–1781, 2011.

C. Doersch, A. Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

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

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings. URL http://proceedings.mlr.press/v9/glorot10a.html.

Jean-Bastien Grill, Florian Strub, Florent Altch'e, C. Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, B. A. Pires, Zhaohan Daniel Guo, M. G. Azar, Bilal Piot, K. Kavukcuoglu, R. Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, abs/2006.07733, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1026–1034, USA, 2015. IEEE Computer Society. ISBN 9781467383912. doi: 10.1109/ICCV.2015.123. URL https://doi.org/10.1109/ICCV.2015.123.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Rong Hu, Brian Mac Namee, and Sarah Jane Delany. Off to a good start: Using clustering to select the initial training set in active learning. In *FLAIRS Conference*, 2010.

Jaeho Kang, Kwang Ryu, and Hyuk-Chul Kwon. Using cluster-based sampling to select initial training set for active learning in text classification. pages 384–388, 05 2004. ISBN 978-3-540-22064-0. doi: 10.1007/978-3-540-24775-3_46.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Y. Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

David Lowell, Zachary Chase Lipton, and Byron C. Wallace. Practical obstacles to deploying active learning. In *EMNLP/IJCNLP*, 2019.

Dmytro Mishkin and Jiri Matas. All you need is a good init. *CoRR*, abs/1511.06422, 2016.

Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. Parting with illusions about deep active learning. *ArXiv*, abs/1912.05361, 2019.

Ali Mottaghi and Serena Yeung. Adversarial representation active learning. *ArXiv*, abs/1912.09720, 2019.

Prateek Munjal, N. Hayat, Munawar Hayat, J. Sourati, and S. Khan. Towards robust and reproducible active learning using neural networks. *ArXiv*, abs/2002.09564, 2020.

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5899–5907, 2017.

Deepak Pathak, Philipp Krähenbühl, J. Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

Soumya Roy, Asim Unmesh, and V. P. Namboodiri. Deep active learning for object detection. In *BMVC*, 2018.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction, 2001.

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

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

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1289–1296, Red Hook, NY, USA, 2007. Curran Associates Inc. ISBN 9781605603520.

Claude E. Shannon and Warren Weaver. *A Mathematical Theory of Communication*. University of Illinois Press, USA, 1963. ISBN 0252725484.

Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and G. Gravier. Rethinking deep active learning: Using unlabeled data at model training. *ArXiv*, abs/1911.08177, 2019.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. *arXiv preprint arXiv:1904.00370*, 2019.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Proceedings of the European Conference on Computer Vision*, 2020.

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