On Initial Pools for Deep Active Learning

Akshay L Chandra[*]
akshaychandra@iith.ac.in

Sai Vikas Desai[*]
cs17mtech11011@iith.ac.in

Chaitanya Devaguptapu[*]
cs19mtech11025@iith.ac.in

Vineeth N Balasubramanian
vineethnb@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. Given the recent successes of learning representations in self-supervised/unsupervised ways, we propose to study if an *intelligently sampled* initial labeled pool can improve deep AL performance. We will investigate the effect of intelligently sampled initial labeled pools, including the use of self-supervised and unsupervised strategies, on deep AL methods. We describe our experimental details, implementation details, datasets, performance metrics as well as planned ablation studies in this proposal. If intelligently sampled initial pools improve AL performance, our work could make a positive contribution to boosting AL performance with no additional annotation, developing datasets with lesser annotation cost in general, and promoting further research in the use of unsupervised learning methods for AL.

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 [1]. 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 [2, 3, 4, 5, 6, 7] have been recently proposed for AL in the context of deep neural network (DNN) models (deep AL). The problem is 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) ML [8]. 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,

[*] Equal Contribution.

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

to the best of our knowledge, the initial pool is always sampled randomly and labeled [2, 3, 4, 5, 6, 9]. This initial pool design strategy has generally worked well for AL in traditional/shallow ML models. However, the success of AL in DNNs has been not 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 [10, 11, 12] 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, the design of the initial labeled pool has received almost no attention in the community, to the best of our knowledge. Considering the tremendous success of self-supervised learning methods in recent years[13, 14, 15, 16, 17, 18, 19], we ask the question if choosing an initial labeled pool in a better manner 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 hypothesise 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 methods started with our initial pool (ALG - **AL** with **G**uided initial pool) will be lower than those of AL methods started with random initial pool (ALR - **AL** with **R**andom initial pool), across AL cycles. However, as AL cycles increase, we expect to see shorter margins of error difference between ALG and ALR as the effect of our initial pool on the model performance could diminish as labeled pool size increases. Studying the use of unsupervised/self-supervised learning in later epochs could be an interesting direction of future work. If ALG outperforms ALR, 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 ALR methods outperform ALG, the community will still have useful insights from 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.* [12] 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.* [11] 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.* [10] comprehensively tested the performance variance of deep AL methods across 25 runs of experiments. They considered various settings such 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.* [20] showed that initializing the target model with the features obtained from self-supervised pretraining gives AL a kickstart in performance. Contemporaneously, Mottaghi & Yeung [7] also used this technique in combination with a GAN based AL method and reported SOTA results on SVHN, CIFAR10, 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 [21, 22, 23] is well understood by the community; however, there have been no such efforts in understanding the importance of good initialization in data space for AL methods.

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.* [24] 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 [5] 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.

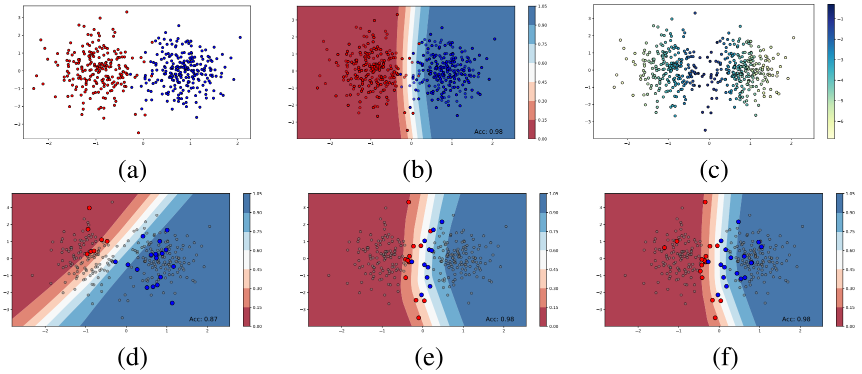
3.1 Pool-based Active Learning Setting

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** to be labeled by an oracle. 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 satisficing 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 ALG methods is lower than those of ALR methods 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.

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



(a) (b) (c)

(d) (e) (f)

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 [9] (f) using Max-Entropy method [25]; Best viewed in color. Labeled instances are emphasized for clarity.

[19, 18, 17, 14, 15, 13] 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 [16] and image rotation prediction [14] tasks. 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, is a hard sample - thus human labeling is needed.* In addition to the above tasks, we will also train a Variational Autoencoder (VAE) [26] 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.

Unsupervised Learning (Clustering) Methods: Sampling bias is the most fundamental challenge posed by AL especially in case of uncertainty based AL methods [27]. 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 looks less and 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 [3, 6] 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 samples 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 [28], albeit in an unsupervised way.

We assume that number of classes to be labeled (K) in the dataset $\mathcal{D}$ is known apriori, which is known in the AL context. 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 [15] 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) [9], Max-Entropy (ME) [25], Min-Margin (MM) [29] & Deep Bayesian AL (DBAL) [2]
- **Diversity Sampling:** Coreset (greedy) [3] & Variational Adversarial AL (VAAL) [6]
- **Query-by-Committee Sampling:** Ensemble with Variation Ratio (ENS-varR) [4] (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 [10], and we will leverage it to study the methods.

4 Implementation Details

Following recent deep AL efforts, we will use MNIST, CIFAR10, CIFAR100 and TinyImageNet (200 classes) [30] datasets in our experiments. We use the AL methods, model architectures, data augmentation schemes and implementation details from Munjal *et al.* [10] 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 [31] across the datasets. For datasets CIFAR10 and CIFAR100, we expect to use learning rate (*lr*), weight decay (*wd*) from [10] - ($lr = 5e^{-4}$, $wd = 5e^{-4}$) 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 [2], [3]. In case of TinyImageNet, before horizontal flipping, we first randomly crop and resize the input to 224 × 224. We will use 18-layer ResNet [32] for all our experiments.

AL Details: As usually done in most AL work, we will initialize L_0 with 5% 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 details due to irregularities between datasets (*e.g.* MNIST is easier to learn compared to TinyImageNet) 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.* [33] 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.

[2] `https://github.com/pytorch/examples/`

[3] `https://github.com/kuangliu/pytorch-cifar`

We will also compare and address the relative performance gain of our clustering methods in case of MNIST and TinyImageNet, two highly contrasting datasets in terms of data and task complexities. We thank the reviewers for suggesting this comparison.

Additionally, we will test the usefulness of our sampling methods on AL for imbalanced data. For this, we will follow Cui *et al.* [34] to simulate a long-tailed distribution of classes on CIFAR100, by following power law.

References

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

[2] 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.

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

[4] 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.

[5] 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.

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

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

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

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

[10] 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.

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

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

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

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

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

[16] 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.

[17] 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.

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

[19] 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.

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

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

[22] 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.

[23] 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.

[24] 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.

[25] Claude E. Shannon and Warren Weaver. *A Mathematical Theory of Communication*. University of Illinois Press, USA, 1963.

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

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

[28] 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.

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

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

[31] 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.

[32] 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.

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

[34] 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.