Reducing Label Effort: Self-Supervised meets Active Learning

Javad Zolfaghari Bengar[1,2] Joost van de Weijer[1,2] Bartlomiej Twardowski[1]
Bogdan Raducanu[1,2]
Computer Vision Center (CVC)[1], Univ. Autònoma of Barcelona (UAB)[2]
{jzolfaghari,joost,btwardowski,bogdan}@cvc.uab.es

Abstract

Active learning is a paradigm aimed at reducing the annotation effort by training the model on actively selected informative and/or representative samples. Another paradigm to reduce the annotation effort is self-training that learns from a large amount of unlabeled data in an unsupervised way and fine-tunes on few labeled samples. Recent developments in self-training have achieved very impressive results rivaling supervised learning on some datasets. The current work focuses on whether the two paradigms can benefit from each other. We studied object recognition datasets including CIFAR10, CIFAR100 and Tiny ImageNet with several labeling budgets for the evaluations. Our experiments reveal that self-training is remarkably more efficient than active learning at reducing the labeling effort, that for a low labeling budget, active learning offers no benefit to self-training, and finally that the combination of active learning and self-training is fruitful when the labeling budget is high. The performance gap between active learning trained either with self-training or from scratch diminishes as we approach to the point where almost half of the dataset is labeled.

1. Introduction

Deep learning methods obtain excellent results on large annotated datasets [29]. However, labeling large amounts of data is labor-intensive and can be very costly. Therefore, the field of active learning explores algorithms that reduce the amount of labeled data that is required. This is achieved by labeling those unlabeled data samples (from the unlabeled data pool) that are considered most useful for the machine learning algorithm. The field of active learning can be roughly divided into two subfields. Informativeness-based methods aim to identify those data samples for which the algorithm is most uncertain [4, 48, 20]. Adding these samples to the labeled data pool is expected to improve the algorithm. Representativeness-based methods aim to label data in such a way that for all unlabeled data there is a 'representative' (defined based on distance in feature space) labeled sample [14, 40]. Active learning methods are typically evaluated by supervised training of the network on only the labeled data pool: the active learning method that obtains the best results, after a number of training cycles with a fixed label budget, is then considered superior.

Self-supervised learning of representation for visual data has seen stunning progress in recent years [6, 7, 8, 19, 22], with some unsupervised methods being able to learn representations that rival those learned supervised. The main progress has come from a recent set of works that learn representations that are invariant with respect to a set of distortions of the input data (such as cropping, applying blur, flipping, etc). In these methods, two distorted versions, called views, of the image are produced. Then the network is trained by enforcing the representations of the two views to be similar. To prevent these networks to converge to a trivial solution different approaches have been developed [19, 50]. The resulting representations are closing the gap with supervised-learned representation. For some downstream applications, such as segmentation and detection, the self-supervised representations even outperform the supervised representations [53].

As discussed, self-supervised learning can learn high-quality features that are almost at par with the features learned by supervised methods. As such it has greatly improved the usefulness of unlabeled data. The standard active learning paradigm trains an algorithm on the labeled data set, and based on the resulting algorithm selects data points that are expected to be most informative for the algorithm in better understanding the problem [41]. In this standard setup, the unlabeled data is not exploited to improve the algorithm. Given the huge performance gains that are reported by applying self-supervised learning, we propose to re-evaluate existing active learning algorithms in this new setting where the unlabeled data is exploited by employing self-supervised learning.

Self-supervised learning and active learning both aim to reduce the label-effort. Based on our experiments we conclude the following:

- In our evaluations on three datasets, Self-training is much more efficient than AL in reducing the labelling effort.
- Self-training + AL substantially outperforms AL methods. However, the performance gap diminishes for large labeling budget (approximately 50% of the dataset in our experiments).
- Based on results of three datasets, Self-training+AL marginally outperforms self-training but only when the labeling budget is high.

In general, our results suggest that self-supervised learning techniques are more efficient than active learning to reduce the label effort. A small additional boost can be obtained from active learning when reaching the high label budget.

Our paper is organized as follows: In section 2 we describe the related work. Next, in section 3 we introduce the proposed framework. Section 4 and 5 present the experimental setup and the evaluations on the datasets we used. Finally, section 6 discusses an interesting finding we observed in our work.

2. Related work

Active learning. Active Learning has been widely studied in various applications such as image classification [18, 16, 9], image retrieval [2], image captioning [11], object detection [55], and regression [15, 25].

Over the past two decades, several strategies have been proposed for sample query, which can be divided in three main categories: informativeness [48, 17, 20, 4, 3], representativeness [39, 40] and hybrid approaches [24, 47]. A comprehensive survey of these frameworks and a detailed discussion can be found in [41].

Among all the aforementioned strategies, the informativeness-based approaches are the most successful ones, with uncertainty being the most used selection criteria used in both bayesian [17] and non-bayesian frameworks [32]. In [17], they obtain uncertainty estimates through multiple forward passes with Monte Carlo Dropout, but it is computationally inefficient for recent large-scale learning as it requires dense dropout layers that drastically slow down the convergence speed. More recently, [1] measures the uncertainty of the model by estimating the expected gradient length. On the other hand, [49, 31] employ a loss module to learn the loss of a target model and select the images based on their output loss.

Representativeness-based methods rely on selecting examples by increasing diversity in a given batch [14]. The Core-set technique [40] selects the samples by minimizing the Euclidian distance between the query data and labeled samples in the feature space. The Core-set technique is shown to be an effective representation learning method, however, its performance is limited by the number of classes in the dataset. Furthermore, Core-set, like other distance-based approaches, are less effective due to feature representation in high-dimensional spaces since p-norms suffer from the curse of dimensionality [13]. In a different direction, [42] uses an adversarial approach for diversity-based sample query, which samples the data points based on the discriminator's output, seen as a selection criteria. Following the same strategy, improved versions have been proposed in [51, 27].

Self-supervised learning. In self-supervised learning, an auxiliary task is introduced. The data for this task should be readily available without the need for any human annotation. The auxiliary task allows to perform unsupervised learning and learn feature representations without the need of labels. Doersch et al. [12] introduce the task of estimating the relative position of image regions. Other examples include coloring gray-scale images [52], inpainting [37], and ranking [33].

In recent years, self-supervised learning has seen a significant performance jump with the introduction of contrastive learning [6], where representations are learned that are invariant with respect to several image distortions. Similar samples are created by augmenting an input image, while dissimilar are chosen by random. This connects to some extent unsupervised setting to previous contrastive methods used in metric learning [21, 44]. To make contrastive training more efficient MoCo method [22] and the improved version [7] use memory bank for learned embeddings what helps with an efficient sampling. This memory is kept in sync with the rest of the network during the training time by using a momentum encoder. Approach named SwAV [5] use online clustering over the embedded samples. In this method negative exemplars are not defined. However, others cluster prototypes can play this role. Even more interesting are methods without any explicit contrastive pairs. BYOL [19] propose asymmetric network by introducing of an additional MLP predictor between two branches' outputs. One of the branch is keep "offline" - updated by a momentum encoder. SimSiam [8] goes even further and presents a simplified solution without a momentum encoder. It comparably good to other methods and does not need a big mini-batch size. A follow up work of BarlowTwins [50] proposes as simple solution as SimSiam with the use of a different loss function - a correlation based one for each pair in current training batch. Here, negatives are implicitly assumed to be in each mini-batch. No asymmetry is used in the network at all, but a bigger embedding size and mini-batches are proffered in comparison to SimSiam.

Previous works that integrated Active Learning and Self-supervised learning include [54, 36]. [54] proposes a query based graph AL method for datasets having structural re-

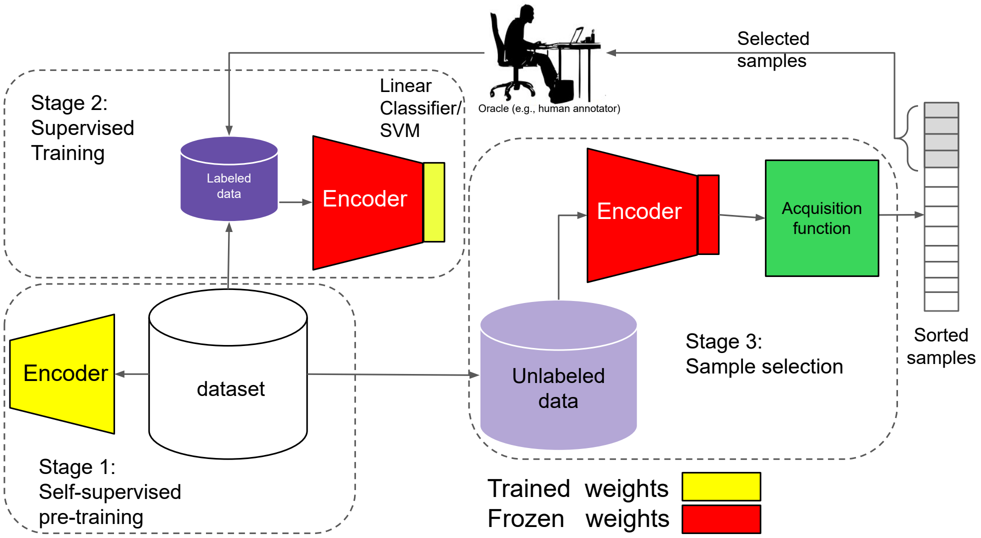

Figure 1. **Overview of active learning framework enhanced by self supervised pre-training.** The framework consists of 3 stages: (i) Self supervised model is trained on the entire dataset. (ii) Given the frozen backbone and few labeled data, a linear classifier or an SVM is fine-tuned on top of the features in supervised way. (iii) Running the model as inference on the unlabeled data and sort the samples from least to highest informative/representative via acquisition function. Finally the top samples are queried to oracle for labeling and added to labeled set. Stages (i) & (ii) are repeated until the total labeling budget finishes.

lationships between the samples coming from few classes. In the context of exploration-driven agent, [36] uses Active Learning and Self-training to learn a policy that allows it to best navigate the environment space.

3. Preliminaries

The main objective of this paper is to evaluate and compare the effectiveness of active learning when combined with recent advances in self-supervised learning. For this purpose we have developed a framework that comprises two parts: self supervised pre-training and active learning (see Figure 1). Primarily, we train the self supervised model as the pretrained model on the unlabeled samples. Next, we use an initial labeled data to finetune a linear classifier on top of pre-trained model. Then we run active learning cycles using the fine-tuned model to select the most informative and/or representative samples and query them for labeling. Hence the original dataset becomes partially labeled. We ablate the self-supervised and active learning components to study their benefits.

We start pretraining our model with SimSiam [8] self-supervised model. The model is based on siamese network trying to maximize the similarity between two augmentations of one image, subject to certain conditions for avoiding collapsing solutions. This enables us to obtain meaningful representations without using negative sample pairs. The rich representations could also potentially help the representative based active learning methods. In the remainder of this section we describe the two components of the experimental framework in detail.

3.1. Active Learning

Given a large pool of unlabeled data $\mathcal{D}_\mathcal{U}$ and a total annotation budget B, the goal is to select b samples in each cycle to be annotated to maximize the performance of a classification model. In general, AL methods proceed sequentially by splitting the budget in several *cycles*. Here we consider the batch-mode variant [41], which annotates b samples per cycle, since this is the only feasible option for CNN training. At the beginning of each cycle, the model is trained on the labeled set of samples $\mathcal{D}_\mathcal{L}$. After training, the model is used to select a new set of samples to be annotated at the end of the cycle via an *acquisition function*. The selected samples are added to the labeled set $\mathcal{D}_L$ for the next cycle and the process is repeated until the annotation budget is spent. The acquisition function is the most crucial component and the main difference between AL methods in the literature. In the experiments we consider several acquisition functions including Informativeness [10] and Representativeness based methods [42, 40].

3.2. Self-supervised Learning

In this section, we shortly introduce self-supervised learning without contrastive sampling and more particularly SimSiam [8], the architecture we employ in this paper.

For a given dataset $\mathcal{D}$, contrastive learning assumes sampling pairs of data points in order to create a good representation. Two main types of pairs are considered: *seman-*

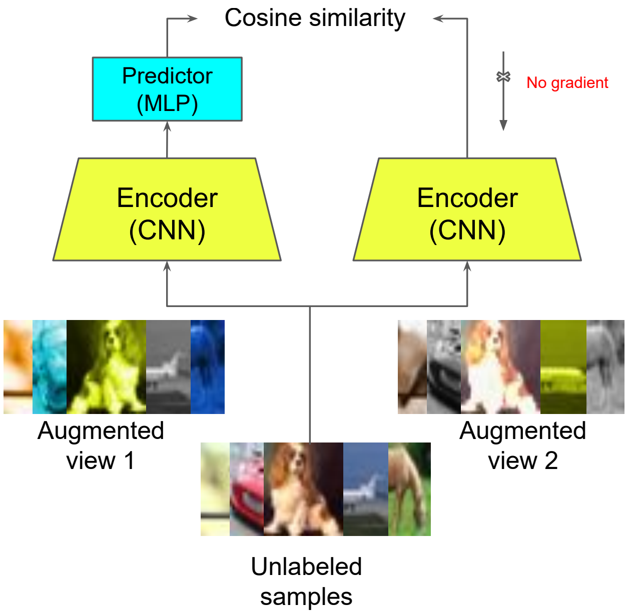

Figure 2. **SimSiam architecture** Two augmented views of one image are processed by the same encoder network (a backbone plus a projection MLP). Then a prediction MLP is applied on one side, and a stop-gradient operation is applied on the other side. The model maximizes the similarity between both sides.

tically similar pairs (x, x^+) – provide an information about some form of close relation of data (based on labeled or unlabeled data); *negative pairs* (x, x^-) – in contrast to positives ones, two non-related samples are given. It is presumed that for a given x, x^- is dissimilar to x^+. Then, contrastive losses [21, 44] learn a new embedding space where a distance between positive pairs is smaller than negatives ones with some margin, e.g. $d(x, x^+) < d(x, x^-) + m$ for triplet loss [44]. That is a core of many metric learning methods [26, 34], where existing labels are used for a semantic similarity check.

Contrastive learning is also often applied in self-supervised learning methods. These methods aim to learn a semantically rich feature representation without the need of any labels. Different augmented views of the same image x form positive samples, while augmentation of different ones provide negatives. This is the base of SimCLR [6] method. However, it's shown that methods without explicit negative sampling prove competitive performance as well, e.g. SimSiam [8] or BYOL [19]. In such methods some additional architecture changes are usually applied, like using asymmetry with an additional predictor network as presented in Figure 2 for SimSiam. The main part is an encoder (CNN based network), learned end-to-end in an asymmetric Siamese architecture, where one branch got an additional predictor (MLP network) which outputs aims to be as close as possible to the other branch. The second branch is not updated in a backward propagation while training. For the similarity function a negative cosine distance is minimized given as:

$$\mathcal{L} = \mathcal{D}(p_1, z_2)/2 + \mathcal{D}(p_2, z_1)/2 \tag{1}$$

$$\mathcal{D}(p_1, z_2) = -\frac{p_1}{\|p_1\|_2} \cdot \frac{z_2}{\|z_2\|_2}, \tag{2}$$

where z_1, z_2 are encoded values respectively for $x1$ and $x2$ – two different augmented views of the same image x. p_1 and p_2 are encoded values additionally passed by a predictor network. There is no contrastive term in this approach, only the similarity is checked and enforced during learning. In SimSiam, besides simplicity, neither negatives mining nor large mini-batches are needed which significantly reduces the GPU requirements. This makes it a good fit for the evaluation proposed in this paper.

4. Experimental Setup

To study the influence of the initial model, various amounts of initial labeled data and budget sizes are evaluated. For the initial labeled set, we considered 1%, 2% and 10% of the entire dataset that are uniformly selected from all classes at random. For one of the datasets we also evaluate 0.1% and 0.2% budget sizes. Before starting the active learning cycles we train the self-supervised model. Then we use the backbone as encoder from SimSiam architecture, freeze the weights and train a linear classifier or SVM on top of the backbone so we only finetune the last layer. At each cycle we start training either from scratch or, in case of self-training, we start from the pretrained self-supervised backbone. We train the model in c cycles until the total budget is exhausted. In each experiment the budget per cycle is equal to initial labeled set.

Datasets. To evaluate various methods, we use CIFAR10 and CIFAR100 [28] datasets with 50K images for training and 10K for testing. CIFAR10 and CIFAR100 have 10 and 100 object categories respectively and an image size of 32×32. To evaluate the scalability of the methods we evaluate on Tiny ImageNet dataset [30] with 90K images for training and 10K for testing. There are 200 object categories in Tiny ImageNet with an image size of 64×64.

Data Augmentation We use different augmentation policies for self supervised pre-training and supervised finetuning. [56] discusses how self-training outperforms normal pre-training in terms of stronger augmentation. For the self-training similar to [8] we used Geometric augmentations [46]: RandomResizedCrop with scale in [0:2; 1:0] and RandomHorizontalFlip. Color augmentation is ColorJitter with $\{$brightness, contrast, saturation, hue$\}$ strength of $\{0.4, 0.4, 0.4, 0.1\}$ with an applying probability of 0.8, and RandomGrayscale with an applying probability of 0.2.

Blurring augmentation [6] has a Gaussian kernel with std in [0:1; 2:0]. For the supervised training we used the conventional RandomResizedCrop with scale [0.08, 1.0] and RandomHorizontalFlip.

Baselines. For the evaluation baselines we compared with Random sampling and several informative and representative-based approaches including Entropy sampling, KCenterGreedy, VAAL and SVM Min Margin. Below we describe the details of the methods we used.

Entropy [10] is an information theory measure that captures the average amount of information contained in the predictive distribution, attaining its maximum value when all classes are equally probable. Entropy sampling selects the most uncertain samples with highest entropy.

As a prominent representative method we evaluate *KCenterGreedy*, which is a greedy approximation of KCenter problem also known as min-max facility location problem [45]. The method selects samples having maximum distance from the nearest labeled samples in the embedding space. We compute the embeddings by running the self-trained model on unlabeled samples.

VAAL [42] is one of state-of-art methods that uses a variational autoencoder to map the distribution of labeled and unlabeled data to a latent space. A binary adversarial classifier is trained to predict if an image belongs to the labeled or the unlabeled pool. The unlabeled images which the discriminator classifies with lowest certainty as belonging to the labeled pool are considered to be the most representative. We used their official code and adapted them into our code to ensure an identical setting. To adapt VAAL for the self-training experiment we initialized and froze the backbone of the task learner.

SVM Min Margin [43] learns a linear SVM on the existing labeled data and chooses the samples that are closest to the decision boundary. To generalize SVM for the multi-class classification problem we adopt it by querying the samples that reside in margin area of decision boundaries.

Implementation details. Our method is implemented in PyTorch [35]. We train Resnet18 [23] that is widely used on CIFAR10 and CIFAR100 datasets. For the self-supervised training, the models are trained with SGD optimizer with momentum 0.9 and base learning rate of 0.03. As in [8] we train models for 800 epochs with batch-size of 512. We use a weight decay of 0.0001 for all parameter layers, including the BN scales and biases, in the SGD optimizer.

Given the pre-trained network, we train a supervised linear classifier on frozen features, which are from ResNet's global average pooling layer. The linear classifier training uses base lr=30 with a cosine decay schedule for 100 epochs, weight decay = 0, momentum=0.9, batch size=256 with SGD optimizer.

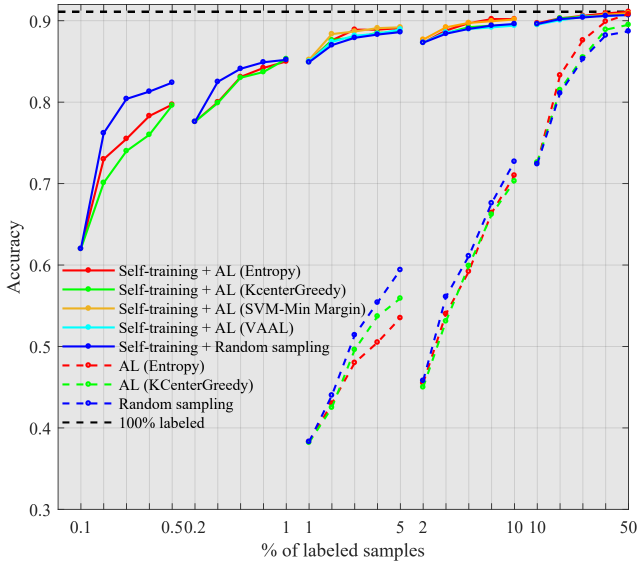

Figure 3. **AL performance on cifar10** performance comparison between the addition of self-training to AL methods (solid lines) and AL methods (dashed lines). The initial and per cycle budget are equal in all the curves.

To implement the SVM for the Min Margin method we used scikit learn python package [38] with linear kernel and set the regularization parameter to 5 in the experiments. To handle the multi-class problem, a one-vs-the-rest classification scheme is chosen.

5. Experiments

To evaluate active learning methods we consider several scenarios in the initial labeled set and budget sizes. For the simplicity we refer to lower than 2% budget sizes as low budget regimes. In this section we inspect the contribution of self-supervised pre-training in active learning.

Performance on CIFAR10. Figure 3 shows active learning results on CIFAR10 dataset. The initial and per cycle budgets are 0.1%, 0.2%,1%, 2% and 10% of labeled data. The evaluated methods are divided into two groups: (i) methods using self-supervised pre-training represented by solid lines. (ii) Methods using models trained from scratch represented by dashed lines. As can be seen, self-training substantially improves all the sampling methods. In particular at the low budget regime, self-training drastically reduces the required labeling. Both types of methods achieve almost the full performance after labeling 50% of data that closes the gap between the self-supervised and supervised methods. The exact numbers are in Table 1. From the active learning perspective, Random sampling outperforms AL methods when the budget is less than 1%. However

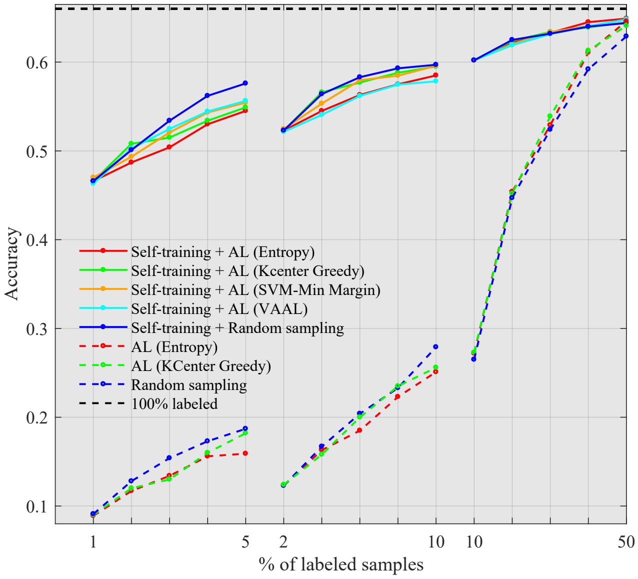

Figure 4. **AL performance on cifar100** performance comparison between the addition of self-training to AL methods (solid lines) and AL methods (dashed lines). The initial and per cycle budget are equal in all the curves.

from 1% budget onward, AL + self-training methods transition to higher performance compared to Random sampling with self-training. For AL methods, trained from scratch, this transition happens after labeling 10% of data. Among AL methods with self-training, Entropy as informativeness method outperforms KCenterGreedy and VAAL. Note that the greatest active learning gain as a result of using self-training occurs after labeling 30% providing 20% less annotation that is equivalent to 10000 less labeling.

Performance on CIFAR100. Figure 4 presents active learning results on CIFAR100 dataset. The three set of curves correspond to three initial and per cycle budget sizes: 1%, 2% and 10%. Solid lines represent AL methods using self-supervised training. While dashed curves correspond to algorithms trained from scratch. As can be seen, self-training dramatically improves the methods without self training. In the low budget regime, self-training significantly reduces the required labeling. While AL methods w/o self-training achieve comparable performance to self-trained counterparts as we approach to 50% labeled data, meaning that the impact of self supervised pre-training diminishes when the budget increases. See Table 1 for detailed numbers. This can also be due to reaching almost the full performance. On CIFAR100, Random sampling outperforms Active learning methods under low budget regardless of using self-training. None of the studied methods foresee a regime where the labeling budget is small, for example, labeling lower than 10%. Among the AL methods with self-training, representative-based methods perform better than Entropy as informative-based in low budget. On CIFAR100, the active learning gain of using self-training appeared almost after labeling 40% of dataset resulting in 10% less annotation that is equivalent to 5000 less labeling.

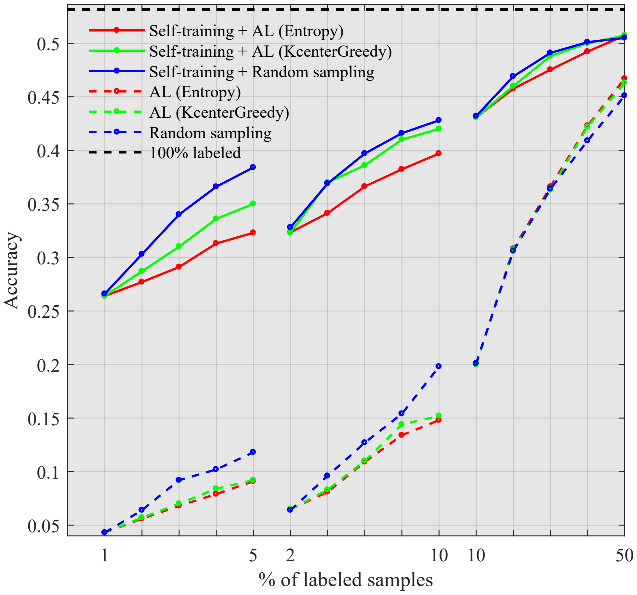

Figure 5. **AL performance on Tiny ImageNet** performance comparison between the addition of self-training to AL methods (solid lines) and AL methods (dashed lines). The initial and per cycle budget are equal in all the curves.

Performance on Tiny ImageNet. Tiny ImageNet is a challenging dataset in terms of diversity of classes. Active learning results on this dataset is presented in Figure 4. Similar to CIFAR100, the three set of curves correspond to 1%, 2% and 10% budget per cycle. Solid lines represent AL methods with self-supervised pre-training and dashed lines correspond to methods trained from scratch. As in other datasets, Self-training drastically reduces the required labeling in low budget scheme. As the labeling increases to 50% AL methods approach the performance of self-trained counterparts. However, unlike CIFAR datasets, AL methods require more than 50% labeling to close the performance gap they have from self-trained counterparts. Among the methods using self training, Random sampling shows superior performance. However, increasing labeled data reduces performance gap from the AL methods. For AL methods w/o self-training, the labeling budget is required to exceed 10% to improve upon Random sampling. In general, active learning fails to perform well under low budget regardless of using self-training. Again AL methods are not designed for low budget regime. Unless the model is trained from scratch with greater than 10% labeling budget, we observe no improvement with the usage of Active learning.

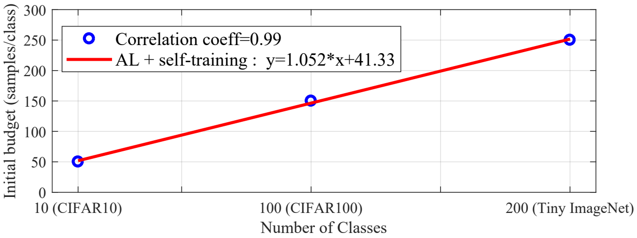

Figure 6. **Correlation between number of samples per class required for AL and number of classes in the datasets.** Above these budgets, AL outperforms Random sampling in the self-supervised setting.

	Methods	Datasets	
		CIFAR10	CIFAR100
AL w/o Self-training	Entropy	0.908	0.646
	KCenterGreedy	0.895	0.641
AL + Self-training	Entropy	0.911	0.649
	SVM Min Margin	0.909	0.644
	VAAL	0.907	0.648
	KCenterGreedy	0.909	0.645

Table 1. **Performance of AL methods with and without Self-training at** 50% **labeling.** For the high labeling budget, the gap between the performances of AL and AL+ Self-training is diminished.

6. Discussion

The experiments in the previous section demonstrated that active learning methods enhanced by self-training do not work well in all budget schemes. However, it might be possible to estimate budgets above which the AL methods outperform Random sampling. Our experiments on three object recognition datasets show that there's a strong correlation (corr. coeff=0.99) between the number of samples per class required for AL and the number of classes in a datasets. Figure 6 presents the thresholds for the budget required for active learning to improve upon Random sampling when uses self-training. This is one interesting finding we observed which can provide a guideline based on the number of classes in a dataset to decide with a certain labeling budget whether it's beneficial to use active learning.

7. Conclusions

This paper analyzed active learning and self supervised approaches independently and unified to investigate how they can benefit from each other. Our experiments demonstrated that self-training is way more efficient than active learning at reducing the labeling effort. Besides, for a low labeling budget, active learning brings no benefit to self-training. Finally, the combination of active learning and self-training is beneficial only when the labeling budget is high. The performance gap between active learning with and without self-training diminishes as we approach to the point where almost half of the dataset is labeled.

References

[1] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019. 2

[2] Björn Barz, Christoph Käding, and Joachim Denzler. Information-theoretic active learning for content-based image retrieval. In *GCPR*, pages 650–666, 2018. 2

[3] Javad Zolfaghari Bengar, Bogdan Raducanu, and Joost van de Weijer. When deep learners change their mind: Learning dynamics for active learning. *arXiv preprint arXiv:2107.14707*, 2021. 2

[4] Wenbin Cai, Ya Zhang, Siyuan Zhou, Wenquan Wang, Chris Ding, and Xiao Gu. Active learning for support vector machines with maximum model change. In *Machine Learning and Knowledge Discovery in Databases*, pages 211–226. Springer, 2014. 1, 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2, 4, 5

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1, 2, 3, 4, 5

[9] Jongwon Choi, Kwang Moo Yi, Jihoon Kim, Jinho Choo, Byoungjip Kim, Jinyeop Chang, Youngjune Gwon, and Hyung Jin Chang. Vab-al: Incorporating class imbalance and difficulty with variational bayes for active learning. In *CVPR2021*, pages 6749–6758, 2021. 2

[10] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995. 3, 5

[11] Yue Deng, KaWai Chen, Yilin Shen, and Hongxia Jin. Adversarial active learning for sequence labeling and generation. In *IJCAI*, pages 4012–4018, 2018. 2

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2

[13] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000. 2

[14] Suyog Dutt Jain and Kristen Grauman. Active image segmentation propagation. In *CVPR*, pages 2864–2873, 2016. 1, 2

[15] A. Freytag, E. Rodner, and J. Denzler. Selecting influential examples: Active learning with expected model output changes. In *ECCV*, pages 562–577, 2014. 2

[16] Weijie Fu, Meng Wang, Shijie Hao, and Xindong Wu. Scalable active learning by approximated error reduction. In *KDD*, pages 1396–1405, 2018. 2

[17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, pages 1183–1192, 2017. 2

[18] E. Gavves, T. E. J. Mensink, T. Tommasi, and T Snoek, C. G. M.and Tuytelaars. Active transfer learning with zero-shot priors: Reusing past datasets for future tasks. In *ICCV*, pages 2731–2739, 2015. 2

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. 1, 2, 4

[20] Yuhong Guo. Active instance sampling via matrix partition. In *NIPS*, pages 1–9, 2010. 1, 2

[21] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2, 4

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, Jun 2016. 5

[24] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *IEEE Trans. on PAMI*, 10(36):1936–1949, 2014. 2

[25] Christoph Käding, Erik Rodner, Alexander Freytag, Oliver Mothes, Björn Barz, and Joachim Denzler. Active learning for regression tasks with expected model output changes. In *BMVC*, pages 1–15, 2018. 2

[26] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019. 4

[27] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *CVPR*, pages 8166–8175, 2021. 2

[28] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. PhD thesis, University of Toronto, 2012. 4

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[30] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7:7, 2015. 4

[31] Minghan Li, Xialei Liu, Joost van de Weijer, and Bogdan Raducanu. Learning to rank for active learning: A listwise approach. In *ICPR*, pages 5587–5594, 2020. 2

[32] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *CVPR*, pages 859–866, 2013. 2

[33] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1862–1878, 2019. 2

[34] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020. 4

[35] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[36] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019. 2, 3

[37] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 5

[39] P.T. Saito, C.T. Suzuki, J.F. Gomes, P.J. de Rezende, and A.X. Falcão. Robust active learning for the diagnosis of parasites. *Pattern Recognition*, 48(11):3572–3583, 2015. 2

[40] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 1, 2, 3

[41] Burr Settles. *Active learning*. Morgan Claypool, 2012. 1, 2, 3

[42] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, pages 5972–5981, 2019. 2, 3, 5

[43] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001. 5

[44] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006. 2, 4

[45] Gert W Wolf. Facility location: concepts, models, algorithms and case studies. *International Journal of Geographical Information Science*, 25(2):331–333, 2011. 5

[46] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 4

[47] Yazhou Yang and Marco Loog. A variance maximization criterion for active learning. *Pattern Recognition*, 78:358–370, 2018. 2

[48] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *IJCV*, 113(2):113–127, 2015. 1, 2

[49] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *CVPR*, pages 93–102, 2019. 2

[50] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021. 1, 2

[51] Beichen Zhang, Liang Li, Shijie Yang, Shuhui Wang, Zheng-Jun Zha, and Qinming Huang. State-relabeling adversarial active learning. In *CVPR*, pages 8756–8765, 2020. 2

[52] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[53] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 1

[54] Yanqiao Zhu, Weizhi Xu, Qiang Liu, and Shu Wu. When contrastive learning meets active learning: A novel graph active learning paradigm with self-supervision. *arXiv preprint arXiv:2010.16091*, 2020. 2

[55] Javad Zolfaghari Bengar, Abel Gonzalez-Garcia, Gabriel Villalonga, Bogdan Raducanu, Hamed Habibi Aghdam, Mikhail Mozerov, Antonio M López, and Joost van de Weijer. Temporal coherence for active learning in videos. In *ICCV Workshops*, 2019. 2

[56] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020. 4