MiSAL: Active Learning for Every Budget

Anonymous authors
Paper under double-blind review

Abstract

In supervised Active Learning (AL), the learner can manipulate the labeled training set by choosing examples to be labeled by an oracle. The size of the labeled set is termed *budget*. Recent years have seen significant progress in this domain in the context of deep active learning. In particular, it has been shown that in general, different families of AL strategies are suitable for high and low budgets. Here we address for the first time the problem of deciding which family of strategies is most suitable for a given budget in a given task. We start from the theoretical analysis of a mixture model, which motivates a computational approach to decide on the most suitable family of methods for the task and budget at hand. We then propose a practical decision algorithm, which determines what family of strategies should be preferred. Using this algorithm, we introduce MiSAL – a mixed strategy active learning algorithm. MiSAL combines AL strategies from different families, resulting in a method that fits all budgets. We support the analysis by an empirical study, showing the superiority of our method when dealing with image datasets.

1 Introduction

In active learning, a learner is allowed to actively interfere with the learning process in order to improve the outcome. Here we consider active learning in the context of the traditional supervised learning framework, in which a learner can influence and shape the construction of the labeled training set. Specifically, we assume that the learner is given an initial set of labeled examples (which could be empty) and a set of unlabeled examples. The learner is then asked to choose a subset of the unlabeled set (termed the active set), up to a fixed budget, and subsequently receive the labels of the selected examples from an oracle. The challenge is to do this in an optimal manner. An effective *active learner* is expected to do better than the random selection of the active set.

The optimal active learning strategy clearly depends on the learner's inductive biases and the nature of the problem. But even when all this is fixed, Hacohen et al. (2022) showed that the optimal strategy still varies, depending on the size of the set of labeled examples (henceforth termed *budget*). Specifically, when the budget is large, methods based on uncertainty-sampling or related criteria are most effective, while typicality-based methods are effective when the budget is small (see Fig. 1).

In practice, given specific data and budget, it is unclear how to determine whether the budget is considered large or small, making it unclear which active learning strategy should be used. We aim to alleviate this problem, by suggesting a practical algorithm to determine which active learning algorithm should be used. Starting from an abstract theoretical analysis in Section 2, we describe an algorithm to decide upon this question. We then derive an active learning strategy that is suitable for

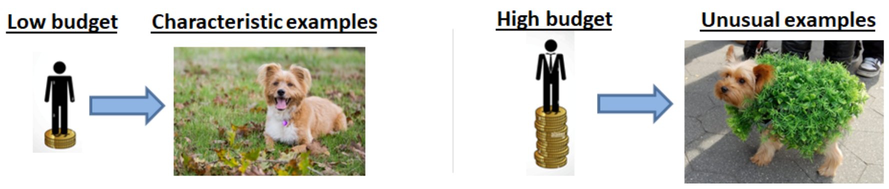


Figure 1: Left: when the budget is low (size of the training set is small), the learner benefits most from seeing characteristic examples. Right: when the budget is high, unusual examples provide the most added value.

all budgets as described in Section 3. The results of an extensive empirical study, showing that the proposed method is beneficial (and often superior) at all budgets, are described in Section 4.

Relation to prior work

By now there is a large body of fundamental work on active learning, see for example the surveys by (Settles, 2009; Schröder & Niekler, 2020). The natural approach, which dominates deep active learning and recent work in particular, aims to identify data whose contribution to the learner achieves maximal added value with respect to what the learner already knows. In the context of supervised learning, the learner may be trained on a fairly large set of already known labeled examples. Afterward, it is allowed to pick unknown examples for human annotation based on the outcome of training. These methods often employ either uncertainty sampling (Lewis & Gale, 1994; Ranganathan et al., 2017; Gissin & Shalev-Shwartz, 2019; Sinha et al., 2019) – seeking examples that the learner is least certain about, diversity sampling (Hu et al., 2010; Elhamifar et al., 2013; Sener & Savarese, 2018; Shui et al., 2020) – seeking examples that effectively span the data space, or some combination of both (Gal et al., 2017; Kirsch et al., 2019; Ash et al., 2020).

But what if not much is known to the learner apriori, and effective training is not possible before the selection of queries? Recently, leveraging advances in unsupervised representation learning (e.g., Van Gansbeke et al., 2020; Chen et al., 2020), another set of active learning methods has emerged (Mahmood et al., 2021; Hacohen et al., 2022; Yehuda et al., 2022). Such works assume a very small (sometimes even empty) set of apriori known labels, and they rely instead on a representation space as obtained by some unsupervised learner. Methods designed to be effective in this domain tend to be qualitatively different from the more traditional methods, seeking examples that can be easily learned rather than seeking confusing examples.

In light of this dichotomy, a new exciting question has emerged when trying to put it all together: in a particular context, what family of methods should be preferred – those designed for high budgets or those designed for low budgets? In particular, can we identify in advance, before we select any query, which family of methods to use? Or maybe a mixture of such methods is to be preferred? To the best of our knowledge, this is the first work that aims to answer this question, developing a practical deep active learning strategy for all budgets. The method is versatile, in that it can incorporate state-of-the-art AL strategies suitable for either the high-budget or low-budget domains.

2 Theoretical Analysis

In the active learning scenario considered here, a learner is given an initial set of labeled examples (which can be empty) and a set of unlabeled examples. The learner is allowed to choose a subset of the unlabeled set, up to a fixed budget, and obtain their labels from an oracle to be used for further training. The selection of examples to be labeled can be done repeatedly, querying in each iteration only part of the total budget. In order to focus the analysis, we consider henceforth a single iteration. In each iteration, the learner is given two sets of examples – one labeled and one unlabeled, and the task is to optimally select a fixed-sized subset of the unlabeled set to be labeled.

In Section 2.1 we introduce some notations and a computational model, which is used in Sections 2.2-2.3 to derive optimal active learning strategies and reveal how they depend on the learning budget.

2.1 Preliminaries

Notations Consider an active learning scenario with an unlabeled set of examples $\mathbb{U}$, a fixed budget $B \in \mathbb{N}$, an active learner $\mathcal{L}$, and an initial set of labeled examples $\mathbb{L}$ of fewer than B examples. The goal of $\mathcal{L}$ is to find an optimal set of examples $\mathbb{A} \subseteq \mathbb{U}$ (referred to as the *active set*), of size $m \equiv |\mathbb{A}| = B - |\mathbb{L}|$, which would maximize the benefit to learner $\mathcal{L}$ when it is to be trained with the labeled set $\mathbb{T} = \mathbb{A} \cup \mathbb{L}$ of size $|\mathbb{T}| = B$.

Active set selection Let $H_{\mathcal{L}}(\mathbb{T})$ denote a hypothesis delivered by learner $\mathcal{L}$ when trained with $\mathbb{T}$, and let $Er(H_{\mathcal{L}}(\mathbb{T}))$ denote its generalization error. Since deep learning training is stochastic, $H_{\mathcal{L}}(\mathbb{T})$ is itself a random variable. Let $E_{\mathcal{L}}(B) = \mathbb{E}[Er(H_{\mathcal{L}}(\mathbb{T}))]$ denote the mean generalization error of $\mathcal{L}$ over all possible hypotheses and all training sets $\mathbb{T}$ of size B.

Model definition and assumptions We adopt the mixture model and assumptions introduced in (Hacohen et al., 2022). The input domain of the mixture model is assumed to be composed of two disjoint regions, R_l and R_h such that $R_l \cup R_h$ is the entire input domain. The mixture model is composed of 2 independent general learners, each learning one of the disjoint regions of the input.

We study the error function of the mixture model on the entire input domain, which depends on the error function of each of the learners in its corresponding region. While the error functions of each learner may be different, we assume for simplicity that their mean generalization error takes on a universal form. In other words, the mean generalization error of each learner is a function that depends only on the size of the training set, up to some constant α which may differ for each region. Formally, the error function of each learner $i \in \{l, h\}$ can be written as $E(\alpha_i B_i)$, where $E : \mathbb{R} \to [0, 1]$ is the universal error function, B_i is the budget in region i, and $\alpha_i \in \mathbb{R}$. We further assume that E has two properties: (i) Efficiency: $E(\alpha B)$ is strictly monotonically decreasing, namely, on average the learner benefits from additional examples. (ii) Realizability: $\lim\limits_{B\to\infty} E(\alpha B) = 0$.

Given the above assumptions, the generalization error of learner $\mathcal{L}$ can be formalized as:

$$E_{\mathcal{L}}(B) = p \cdot E(\alpha_l B_l) + (1-p) \cdot E(\alpha_h B_h). \tag{1}$$

Above p denotes the probability of region R_l. We assume w.l.o.g. that $\alpha_l = 1$ and $\alpha_h \equiv \alpha > 0$. Without loss of generality, we assume that R_l requires fewer examples to be adequately learned than R_h, which implies that $\alpha < \frac{p}{1-p}$. With this assumption, the prior art shows that it is beneficial to sample from the easier-to-learn region R_l when the budget is low, and sample from the harder-to-learn region R_h when the budget is high. Within this framework, the exact transition point between the low and the high budgets depends on the specific parameters of the problem at hand.

2.2 Optimal mixed strategy for query selection

We now analyze active learning strategies using the mixture model defined in Section 2.1. In this model, there exist two pure query selection strategies, denoted $\mathcal{S}_l$ and $\mathcal{S}_h$. $\mathcal{S}_l$ samples points from R_l, and is beneficial for low budgets. $\mathcal{S}_h$ samples points from R_h, and is beneficial for high budgets. Given a fixed budget B, we analyze the family of mixed strategies – each defined by $q \in [0, 1]$, in which qB points are sampled using $\mathcal{S}_l$ and $(1-q)B$ points are sampled using $\mathcal{S}_h$. We seek the optimal mixed strategy denoted $\hat{q}$, which delivers a labeled set of size B with the lowest mean error.

First, we note that the mean generalization error of mixed strategy q is:

$$E_{\mathcal{L}}(B, q) = p \cdot E(qB) + (1-p) \cdot E(\alpha(1-q)B). \tag{2}$$

Since E is differentiable by assumption, we can find the optimum at $\hat{q}$ by differentiating (2) (see derivation in App. A), to obtain:

$$\frac{E^{'}\left(\hat{q}B\right)}{E^{'}\left(\alpha\left(1-\hat{q}\right)B\right)} = \frac{\alpha\left(1-p\right)}{p}. \tag{3}$$

If (3) has a solution for $\hat{q}$ and if this solution is unique and minimal, it defines an optimal mixed strategy $\hat{q}_E(B, p, \alpha)$. $\forall B$ such that $\hat{q}_E(B, p, \alpha) = p$, this optimal mixed strategy is equivalent to random selection from the input domain[1]. We denote such budgets by B_{equiv}, where from (3)

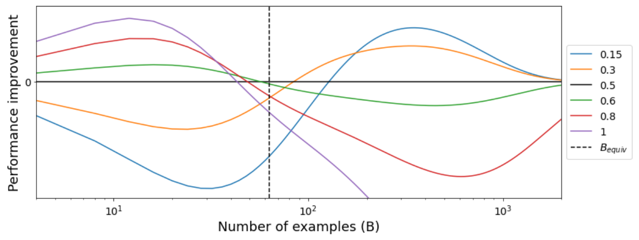
$$\frac{E^{'}\left(pB_{equiv}\right)}{E^{'}\left(\alpha\left(1-p\right)B_{equiv}\right)} = \frac{\alpha\left(1-p\right)}{p}.$$

By definition, when the budget size is B_{equiv}, random query selection is optimal.

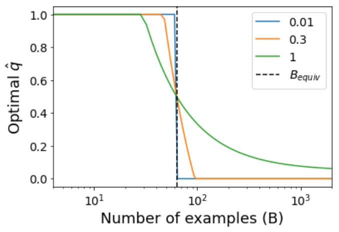
In the analysis below, we assume for simplicity[2] that $\hat{q}_E(B, p, \alpha)$ is non-increasing $\forall B$ and strictly monotonic at B_{equiv}, implying that B_{equiv} is unique. A visualization of this case can be seen in Fig. 2, where we plot the closed form solution of $\hat{q}_a(B, p, \alpha)$ with an exponential universal error function $E(x) = e^{-ax}$.

[1]Identity (in probability) is achieved when $B \to \infty$.

[2]Otherwise, the analysis is to be repeated in each such region of $\hat{q}_E(B, p, \alpha)$.



(a) Accuracy gain of mixed strategies q, where mixture coefficient q is indicated in the legend.



(b) Optimal $\hat{q}_a(B, p, \alpha)$.

Figure 2: Visualization of $E_{\mathcal{L}}(B, q)$, where training set of size B is selected for different values of q as indicated in the legend. The parameters are $E(x) = e^{-ax}$, $a = 0.1$, $p = \frac{1}{2}$ and $\alpha = 0.05$. (a) A plot of accuracy gain when using strategy q as compared to random query selection: $E(p) - E(q)$, as a function of budget B. Since $p = \frac{1}{2}$, the plot corresponding to $q = \frac{1}{2}$ is always 0. (b) Plots of $\hat{q}$ as a function of B. B_{equiv}, which corresponds to $\hat{q} = \frac{1}{2}$, is indicated by a vertical dashed line. Each plot corresponds to a different fraction $\frac{|\mathbb{A}|}{B}$ (see legend).

2.3 Optimal selection strategy: dependence on the size of the active set

In the active learning scenario as defined in Section 2.1 we have $\mathbb{T} = \mathbb{L} \cup \mathbb{A}$, where set $\mathbb{L}$ is fixed and the optimal selection of set $\mathbb{A}$ is sought. Let m denote the size of the required set, namely, $|\mathbb{A}| = m$ and $|\mathbb{L}| = B - m$. Not knowing anything about the origin of $\mathbb{L}$, we further assume that it has been selected randomly from the input domain, which is equivalent to the mixed strategy $q = p$.

In this case, the selection of set $\mathbb{T}$ is more naturally conceived as a mixture between 3 strategies $\{\mathcal{S}_r, \mathcal{S}_l, \mathcal{S}_h\}$, where strategy $\mathcal{S}_r$ randomly selects data from the input domain. This is captured by the notation $\mathcal{S}(r_r, r_l, r_h)$, where (r_r, r_l, r_h) in the 3-simplex, for a query selection strategy that selects $r_r B$ points with $\mathcal{S}_r$, $r_l B$ points with $\mathcal{S}_l$, and $r_h B$ points with $\mathcal{S}_h$. Note that this notation is unique only if either $r_l = 0$ or $r_h = 0$ (or both).

Optimal mixed strategy The unique optimal strategy, which corresponds to mixture $\hat{q} \equiv \hat{q}_E(B, p, \alpha)$, is captured by this notation as:

$$\hat{S}(B, p, \alpha) = \begin{cases} \mathcal{S}(1 - \hat{r}, \hat{r}, 0) & \hat{q} > p \implies B < B_{equiv} \\ \mathcal{S}(1, 0, 0) & \hat{q} = p \implies B = B_{equiv} \\ \mathcal{S}(1 - \hat{r}, 0, \hat{r}) & \hat{q} < p \implies B > B_{equiv} \end{cases} \tag{4}$$

Above $\hat{r}$ is obtained by noting that $\hat{r} + (1 - \hat{r})p = \hat{q}$, which implies that

$$\hat{r} = \begin{cases} \frac{1}{1-p}(\hat{q} - p) & \hat{q} > p \\ \frac{1}{1-p}(p - \hat{q}) & \hat{q} < p \end{cases} \tag{5}$$

Achievable optimal mixed strategy Are all optimal mixed strategies feasible? Since the size of set $\mathbb{A}$ is fixed at m, we can deduce from (5), and the definition of vector (r_r, r_l, r_h), that strategy $\hat{r}$ is *not feasible* whenever one of the following two conditions holds

$$\hat{r}B = \frac{(\hat{q} - p)}{1 - p} B > m \implies \hat{q} > p + \frac{m(1 - p)}{B} > p \implies B < B_{low} \leq B_{equiv}$$

or (6)

$$\hat{r}B = \frac{(p - \hat{q})}{1 - p} B > m \implies \hat{q} < p - \frac{m(1 - p)}{B} < p \implies B > B_{high} \geq B_{equiv}$$

Above, B_{low} is defined to be the solution of (3) when inserting $\hat{q} \to p + \frac{m(1-p)}{B}$, and B_{high} is the solution of (3) when inserting $\hat{q} \to p - \frac{m(1-p)}{B}$ (see derivation of B_{low} and B_{high} in App. A).

We may conclude that the optimal strategy (4) that is also achievable is:

$$\hat{S}(B,p,\alpha)=\begin{cases} S(0,1,0) & B<B_{low} \\ S(1-\hat{r},\hat{r},0) & B_{low}\le B<B_{equiv} \\ S(1,0,0) & B=B_{equiv} \\ S(1-\hat{r},0,\hat{r}) & B_{equiv}<B\le B_{high} \\ S(0,0,1) & B_{high}<B \end{cases} \tag{7}$$

Comparing (7) to (4), we note that at low budgets ($B < B_{low}$) one selects the active set with pure strategy $\mathcal{S}_l$ since more than m points (the size of the active set) are missing from region R_l to achieve optimal performance. At high budgets ($B > B_{high}$) one selects the active set with pure strategy $\mathcal{S}_h$ since more than m points are missing from region R_h to achieve optimal performance.

Achievable optimal mixed strategy: incremental selection Inspecting the definition of the transition points in the optimal achievable strategy, defined in (6), we observe that $\lim_{m\to 0} B_{low} = \lim_{m\to 0} B_{high} = B_{equiv}$. In other words, if the active set $\mathbb{A}$ is small enough, the achievable optimal mixed strategy can be effectively approximated by the following strategy

$$\tilde{S}(B,p,\alpha)=\begin{cases} S(0,1,0) & B<B_{low} \\ S(1,0,0) & B_{low}\le B\le B_{high} \\ S(0,0,1) & B>B_{high} \end{cases} \tag{8}$$

In (8), if the budget size is smaller than B_{low} then the active set is sampled from R_l, and if the budget size is larger than B_{high} then the active set is sampled from R_h. Otherwise, the active set is randomly sampled from all the data. The deviation of (8) from the optimal strategy (7) lies in a segment of budget values whose size tends to 0 as m decreases (see visualization in Fig. 2b).

If m is not sufficiently small to justify the use of strategy (8), query selection may be done incrementally: First, m is to be divided to smaller segments of length m', where m' is small enough to justify the use of (8). Second, strategy (8) is used repeatedly $\frac{m}{m'}$ times, to build the set $\mathbb{A}$ iteratively. While taking longer to run, this iterative strategy can be implemented more easily and more robustly than strategy (7), since it does not require the evaluation of the elusive mixture coefficient $\hat{r}$.

Further support for the use of strategy (8) can be derived from our empirical results (see Section 2.4 below). In our experiments, the difference in the mean generalization error between strategies (7) and (8) is marginal even when the truly optimal $\hat{r}$ is manually handed to the algorithm. This supports the adaptation of strategy (8) in practice with a small number of repetitions.

2.4 VALIDATION AND VISUALIZATION OF THEORETICAL RESULTS

Visualization In Fig. 3, we plot the error of the strategies defined in (7) and (8), using the same exponential example as in Fig. 2. The orange curve represents a fairly large active set, where m (the size of active set $\mathbb{A}$) is equal to 30% of budget B, while the blue curve represents a very small active set, where m is equal to 1% of the budget B. We can see that the smaller m is, the smaller the gap between the optimal strategy in (7) and the achievable strategy in (8) becomes.

Fig.2b shows plots of the optimal mixture coefficient $\hat{q}$ as a function of budget size B, when considering different values for the active set size m, including 1%, 30% and 100% of B. We see that as the theory predicts, the smaller m is, the more step-like the optimal $\hat{q}$ becomes, suggesting that in most cases, the entire active set $\mathbb{A}$ should be sampled from the same strategy.

Validation Our analysis uses a mixture model of idealized general learners. We now validate that similar phenomena occur in practice when training deep networks on different computer vision tasks.

As our choice for pure strategies $\mathcal{S}_l$ and $\mathcal{S}_h$, we use *TypiClust* and *margin* respectively (see details in Section 4.1). Fig. 4 shows results when training $N = 20$ networks using mixed strategies $S(1-r, 0, r)$ and $S(1-r, r, 0)$ on CIFAR-10 and CIFAR-100. We compare the mean performance of each strategy to the performance of $N = 20$ networks trained with the random query selection strategy $S(1, 0, 0)$. We note that other choices for $\mathcal{S}_l$ and $\mathcal{S}_h$ yield similar qualitative results, as can be seen in Tables 1-2.

Inspecting these results, we find similar trends to those shown in the theoretical analysis. The most beneficial value of mixture coefficient r decreases as the budget increases until a certain transition

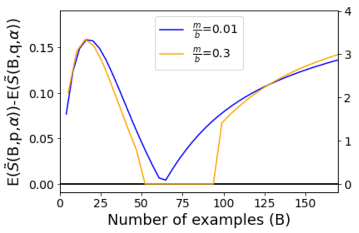
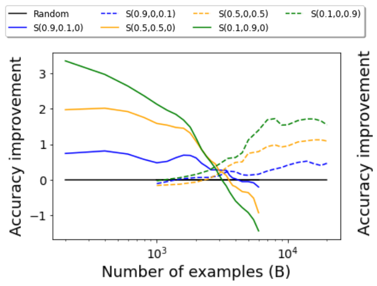
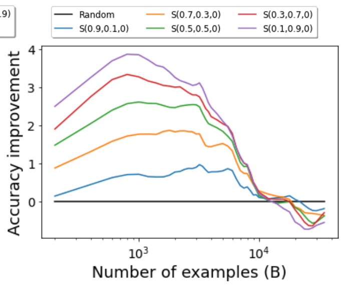


Figure 3: Visualization of strategy (8) for $E(x) = e^{-ax}$ (see Fig. 2). We plot its gain in error reduction as compared to random query selection. Two cases are shown, with small $\frac{m}{B} = 0.01$ and large $\frac{m}{B} = 0.3$ active sets. The smaller the active set is, the closer the performance is to optimal strategy (7).



(a) CIFAR-10



(b) CIFAR-100

Figure 4: Empirical validation of the theoretical results. We plot the accuracy gains of $N = 20$ networks trained using strategy (8), compared to using no active learning at all. We see that as predicted by the theoretical analysis, the most beneficial mixture coefficient r increases as the difference $|B - B_{equiv}|$ increases.

point (corresponding to B_{equiv}). From this transition point onward, the bigger the budget is, the more beneficial it is to select additional examples by one of the high-budget strategies. In fact, as in the theoretical analysis, when the budget is low it is beneficial to use a pure low-budget strategy, when the budget is high it is beneficial to use a pure high-budget strategy, and the transition area in between (corresponding to the segment $B_{low} \leq B \leq B_{high}$) is typically rather short (see Figs. 4 and 5).

3 MISAL: MIXED STRATEGY FOR ACTIVE LEARNING

Following the theoretical discussion in Section 2.3, our method adopts the *achievable optimal mixed strategy* as prescribed in (8). The method involves two main steps. In the first step, a strategy beneficial for low budgets and another beneficial for high budgets is chosen. For each strategy, we evaluate whether it is more effective than random sampling at the given budget B, and estimate its added generalization accuracy for the given query size m as explained in Section 3.1. This step determines the budget's actual domain: (i) low ($B \leq B_{low}$), (ii) transition ($B_{low} \leq B \leq B_{high}$), or (iii) high ($B \geq B_{high}$). In the second step, a strategy is chosen from among the most competitive ones in the relevant domain, and is subsequently used to select m points for active set $\mathbb{A}$.

3.1 DECIDING ON THE SUITABLE BUDGET REGIME

Let $\mathcal{S}_r$ denote the strategy that selects query points randomly. Given two strategies $\mathcal{S}_l$ and $\mathcal{S}_h$ and given budget B, the method outlined in Alg. 1 below aims to determine (separately for each strategy) whether its expected performance is better than random query selection, or not. In other words, we need to determine whether m additional points, as chosen by either $\mathcal{S}_l$ or $\mathcal{S}_h$ respectively, reduce the mean generalization error more significantly than m randomly selected points.

Unfortunately, this test cannot be decided directly without additional labels. Instead, our method computes the result of a surrogate test, where a small set of points as chosen by each respective strategy is removed from the labeled set. It then compares the reduction in mean generalization error to the removal of randomly chosen points.

The proposed surrogate test raises another problem: in most active learning strategies, in particular those suitable for high budgets, the outcome relies on a learner that is trained on the labeled set $\mathbb{L}$ and is thus exposed to the points that are to be removed. As a result, the test is strongly biased to underestimate the cost of removing known points, as also seen in our empirical study (see Fig. 6). Instead, in order to achieve an unbiased surrogate test, we restrict the choice of active learning strategies to methods that rely only on the unlabeled set $\mathbb{U}$. We denote these methods by $\mathcal{S}'_l$ and $\mathcal{S}'_h$.

The final method can now be summarized as follows (see Alg. 1): We begin by considering the unlabeled pool $\mathbb{U}$, and obtain an effective representation for all the data using a self-supervised learning task. In the feature space thus defined, we create three subsets of $\mathbb{L}$: $data_l$, $data_h$ and $data_r$. Each subset is obtained by removing a small number of examples (but not less than 1) from each

Algorithm 1 Select active learning regime, low budget or high budget

```
Input: Unlabeled pool $\mathbb{U}$, Labeled pool $\mathbb{L}$, Budget $B$, learner $\mathcal{L}$, $\epsilon$, 2 Strategies: $\mathcal{S}'_l, \mathcal{S}'_h$
Output: Decision variable $S$
features ← features of data $\mathbb{L}$ obtained by a self-supervised task trained on $\mathbb{U}$
$k \leftarrow$ # of classes,   $c \leftarrow \max\{\lfloor \frac{\epsilon}{k} \rfloor, 1\}$
$data_l,\ data_h \leftarrow$ remove $c$ examples per class as chosen by $S_l$, $S_h$ respectively
$data_r \leftarrow$ remove $c$ examples per class randomly
$acc_l,\ acc_h,\ acc_r \leftarrow$ accuracy of $\mathcal{L}$ trained on $data_l,\ data_h,\ data_r$ respectively
if $acc_l < acc_r$ and $acc_l \leq acc_h$ then
    $S \leftarrow$ '$low$'
else if $acc_h < acc_r$ and $acc_h \leq acc_l$ then
    $S \leftarrow$ '$high$'
else
    $S \leftarrow$ '$rand$'
end if
return $S$
```

class as chosen by $\mathcal{S}'_l$, $\mathcal{S}'_h$, and $\mathcal{S}_r$ respectively. We then train the learner on each of these datasets (this is repeated multiple times for $\mathcal{S}_r$). Finally, we compute the accuracy of each method using cross-validation, using 1% of the training data as a validation set in multiple repetitions. We choose the strategy whose accuracy is the lowest – this is the strategy that is most critical for learning, hence removing examples according to it decreases the performance the most.

3.2 Mixed active learning strategy

The active learning strategy for query selection is described below in Alg. 2. First, it selects two active learning strategies, $\mathcal{S}'_l$, and $\mathcal{S}'_h$, which are known to be beneficial in the low and high budget regimes respectively, and which additionally rely in their computation only on the unlabeled set $\mathbb{U}$. Together with the data, they are given as input to Alg. 1, which in turn returns a decision, indicating what regime best describes the given budget B. In its second step, Alg. 2 selects two active learning strategies, $\mathcal{S}_l$, and $\mathcal{S}_h$, which are known to be beneficial in the low and high budget regimes respectively, with no additional restrictions. This allows the selection of strategies more competitive than $\mathcal{S}'_l$ and $\mathcal{S}'_h$. The additional flexibility is especially important in the high budget regime, where the most competitive strategies typically rely in their computation on all the data, including both $\mathbb{L}$ and $\mathbb{U}$. Alg. 2 then uses the preferred AL strategy ($\mathcal{S}_l$, $\mathcal{S}_h$ or $\mathcal{S}_r$) to select the active set $\mathbb{A}$.

Algorithm 2 Mixed-Strategy Active Learning (MiSAL)

```
Input: Unlabeled pool $\mathbb{U}$, Labeled pool $\mathbb{L}$, Budget $B$, learner $\mathcal{L}$
Output: active set $\mathbb{A}$ of $B - |\mathbb{L}|$ examples
$m \leftarrow B - |\mathbb{L}|$
$\mathcal{S}'_l, \mathcal{S}'_h \leftarrow$ competitive AL methods that rely only on $\mathbb{U}$
$S \leftarrow$ output of Alg. 1 given $(\mathbb{U}, \mathbb{L}, B, \mathcal{L}, m, \mathcal{S}'_l, \mathcal{S}'_h)$
$\mathcal{S}_l, \mathcal{S}_h \leftarrow$ competitive AL methods (unrestricted)
if $S ==$ '$low$' then
    $\mathbb{A} \leftarrow m$ points selected by $S_l$ from $\mathbb{U}$
else if $S ==$ '$high$' then
    $\mathbb{A} \leftarrow m$ points selected by $S_h$ from $\mathbb{U}$
else
    $\mathbb{A} \leftarrow m$ random points from $\mathbb{U}$
end if
return $\mathbb{A}$
```

The use of $\mathcal{S}'_l$ and $\mathcal{S}'_h$ in the call to Alg. 1, rather than $\mathcal{S}_l$ and $\mathcal{S}_h$, is a necessary evil as explained in Section 3.1. The rationale for expecting our method to succeed lies in the theoretical analysis presented in Section 2, which supports the hypothesis that the transition points B_{low} and B_{high} may be universal, or approximately so, across pure strategies. Our empirical results, reported in Section 4, support this supposition.

4 Empirical Results

We now describe the empirical results of our integrated active learning strategy, where Alg. 2 is used to generate the active set. After labels are obtained for the active set, the training of the deep model proceeds as is customary in deep supervised learning, using all the available labels in $\mathbb{A} \cup \mathbb{L}$.

4.1 Methodology

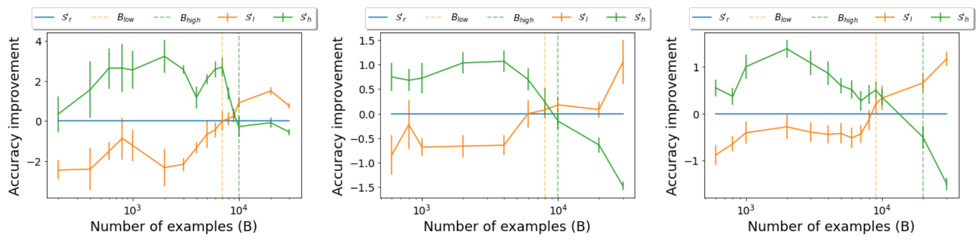
Our experimental framework is based on the code of (Munjal et al., 2020), which allows for the comparison of different active learning strategies in a robust and fair way. While the architecture trained by this framework does not achieve state-of-the-art results on CIFAR and ImageNet, it makes it possible to compare many active learning methods in a competitive environment, while using an architecture with which these strategies have been shown to be beneficial. In the following experiments, we trained ResNet-18 (He et al., 2015) on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) and ImageNet-50 – a subset of ImageNet (Deng et al., 2009) containing 50 classes as done in (Van Gansbeke et al., 2020). Hyper-parameters are the same as in Munjal et al. (2020), see App. B.

As our choice of competitive pure active learning strategies that use only the unlabeled set $\mathbb{U}$ for training (see Alg. 2), we choose *TypiClust* (Hacohen et al., 2022) for $\mathcal{S}'_l$ and *inverse TypiClust* for $\mathcal{S}'_h$. In the latter strategy, the most atypical examples are selected; we note that *inverse TypiClust* is an effective strategy for high budgets, see App. C.1. As our unrestricted choice of competitive pure AL strategies, we choose *ProbCover* (Yehuda et al., 2022) for $\mathcal{S}_l$ and *BADGE* (Ash et al., 2020) for $\mathcal{S}_h$. Other choices yield similar patterns of improvement, as can be verified from Tables 1-2.

In the experiments below, we use several active learning strategies, including *Min margin*, *Max entropy*, *Least confidence*, *DBAL* (Gal et al., 2017), *CoreSet* (Sener & Savarese, 2018), *BALD* (Kirsch et al., 2019), *BADGE* (Ash et al., 2020), *TypiClust* (Hacohen et al., 2022) and *ProbCover* (Yehuda et al., 2022). When available, we use for each strategy the code provided in (Munjal et al., 2020). For low-budget strategies, which are not implemented in (Munjal et al., 2020), we used the available code provided in the git of each paper. When using Alg. 2, the underline feature space is SimCLR. Other choices of feature spaces yield similar results, see the comparison in App. 8.

4.2 Evaluating Alg. 1 in isolation

We start by isolating the strategy selection test as describe in Alg. 1 in Section 3.1. In order to generate the 3 subsets of labeled examples $data_l$, $data_h$ and $data_r$, we remove 5% of the labeled data (but not less than 1 datapoint per class). Results are shown in Fig. 5. We can see that in the low-budget regime, removing examples according to $\mathcal{S}'_l$ yields worse performance as compared to the removal of random examples, while better performance is seen in the high-budget regime. The opposite behavior is seen when removing examples according to the high-budget strategy $\mathcal{S}'_h$.



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet-50

Figure 5: Accuracy gain when using $\mathcal{S}'_l$ to select points for removal as compared to random selection (orange), or $\mathcal{S}'_h$ to select points for removal (green). Negative gain implies that the strategy is beneficial, and vice versa.

4.3 Optimal strategy: results

Tables 1-2 show the performance of our proposed method in comparison with the performance of the other baselines. In all these experiments, the integrated strategy is successful in its identification of a suitable budget regime. As a result, it works well both in the low and high-budget regimes, matching or surpassing both the low and high-budget strategies at all budgets. As MiSAL chooses an active learning strategy dynamically for each budget, any state-of-the-art improvements for either low or high budgets AL strategies would also reflect an improvement of MiSAL as well.

Table 1: Mean accuracy and standard error of $N = 10$ networks trained on CIFAR-10 and CIFAR-100, using various budgets and active learning strategies. In each dataset, we display results for 3 budget choices: one smaller than B_{low} (left column), one between B_{low} and B_{high} (middle column), and one larger than B_{high} (right column). We highlight in boldface the best result in each column, and additionally all the results that lie within its interval of confidence (the standard error bar). While most strategies are effective only in the low or the high budgets, MiSAL is effective in both regimes. As predicted, between B_{low} and B_{high}, most AL strategies do not significantly outperform random query selection.

	CIFAR-10			**CIFAR-100**		
Budget ($\mathbb{L} + \mathbb{A}$)	100+100	7k+1k	25k+5k	100+100	9k+1k	30k+7k
random	31.8 ± 0.3	$\mathbf{76 \pm 0.3}$	87.2 ± 0.2	5.3 ± 0.2	$\mathbf{39.9 \pm 0.3}$	60.7 ± 0.3
TypiClust	34.1 ± 0.4	75.7 ± 0.3	87.1 ± 0.2	7.1 ± 0.1	$\mathbf{39.7 \pm 0.4}$	60.4 ± 0.2
BADGE	31.3 ± 0.4	$\mathbf{76.5 \pm 0.4}$	$\mathbf{88.1 \pm 0.1}$	5.3 ± 0.2	$\mathbf{39.5 \pm 0.3}$	$\mathbf{61.9 \pm 0.1}$
DBAL	30.2 ± 0.3	$\mathbf{76.4 \pm 0.4}$	87.8 ± 0.1	4.6 ± 0.2	38.9 ± 0.4	61.5 ± 0.2
BALD	30.8 ± 0.3	$\mathbf{76.3 \pm 0.2}$	$\mathbf{88 \pm 0.2}$	4.9 ± 0.2	$\mathbf{39.8 \pm 0.5}$	61.5 ± 0.2
CoreSet	29.4 ± 0.4	$\mathbf{75.8 \pm 0.3}$	87.7 ± 0.2	5.6 ± 0.4	$\mathbf{39.1} \pm 0.3$	61.4 ± 0.2
ProbCover	$\mathbf{35.1 \pm 0.3}$	$\mathbf{76.1 \pm 0.3}$	87.1 ± 0.1	$\mathbf{8.2 \pm 0.1}$	$\mathbf{40 \pm 0.4}$	61.4 ± 0.3
Min Margin	30.7 ± 0.4	71.1 ± 0.2	$\mathbf{87.9 \pm 0.2}$	5.2 ± 0.1	39.2 ± 0.2	61.6 ± 0.3
Max Entropy	30.2 ± 0.3	$\mathbf{76.1 \pm 0.3}$	$\mathbf{87.8 \pm 0.2}$	4.9 ± 0.2	39 ± 0.2	61.6 ± 0.2
Least Confidence	29.7 ± 0.2	$\mathbf{76.1 \pm 0.3}$	$\mathbf{88.1 \pm 0.2}$	4.7 ± 0.4	38.9 ± 0.4	61.5 ± 0.2
MiSAL	$\mathbf{35.1 \pm 0.3}$	$\mathbf{76 \pm 0.3}$	$\mathbf{88.1 \pm 0.1}$	$\mathbf{8.2 \pm 0.1}$	$\mathbf{39.9 \pm 0.3}$	$\mathbf{61.9 \pm 0.1}$

Table 2: Same as Table 1, using ImageNet 50.

	ImageNet-50		
B ($\mathbb{L} + \mathbb{A}$)	100+100	7k+1k	25k+5k
random	9.3 ± 0.2	$\mathbf{61.8 \pm 0.4}$	79.8 ± 0.2
TypiClust	$\mathbf{11.3 \pm 0.3}$	$\mathbf{61.8 \pm 0.5}$	80.1 ± 0.2
BADGE	9.4 ± 0.2	$\mathbf{61.3 \pm 0.5}$	$\mathbf{80.8 \pm 0.2}$
DBAL	9 ± 0.4	$\mathbf{61.1 \pm 0.6}$	80.1 ± 0.2
BALD	9.4 ± 0.4	$\mathbf{61.7 \pm 0.3}$	$\mathbf{80.7 \pm 0.1}$
CoreSet	8.6 ± 0.3	$\mathbf{61.7 \pm 0.2}$	$\mathbf{80.7 \pm 0.3}$
ProbCover	$\mathbf{11.4 \pm 0.6}$	$\mathbf{61.6 \pm 0.8}$	77.7 ± 0.3
Min Margin	9.8 ± 0.2	$\mathbf{61.2 \pm 0.5}$	80.4 ± 0.1
Max Entropy	8.9 ± 0.2	$\mathbf{61.8 \pm 0.4}$	80.1 ± 0.1
Least Conf.	8.9 ± 0.1	$\mathbf{61.5 \pm 0.4}$	79.6 ± 0.5
MiSAL	$\mathbf{11.4 \pm 0.6}$	$\mathbf{61.8 \pm 0.4}$	$\mathbf{80.8 \pm 0.2}$

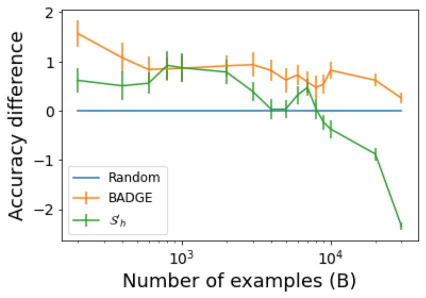


Figure 6: Similarly to Fig. 5a, comparing $\mathcal{S}'_h$ with BADGE as the strategy for the removal of examples. Unlike the original $\mathcal{S}'_h$ (green), BADGE (orange) shows no transition point.

Why should the choice of AL strategies in Alg. 1 be restricted? For a given task and a given budget, Alg. 1 is designed to identify the suitable family of AL strategies, whether low or high budget, using a respective choice of $\mathcal{S}'_l$ and $\mathcal{S}'_h$. To achieve the results reported above, we use *TypiClust* for $\mathcal{S}'_l$ and *inverse TypiClust* for $\mathcal{S}'_h$. In Section 3.1 we discuss why we do not use to this end the subsequent selection of the most competitive strategies $\mathcal{S}_l$ and $\mathcal{S}_h$, arguing that the selection must be restricted to AL strategies that do not rely on the labeled set $\mathbb{L}$. We now demonstrate what happens when the selection is not restricted in this manner, and in particular if $\mathcal{S}'_h$ is chosen to be a competitive AL strategy that relies on the labeled set $\mathbb{L}$ for its successful outcome. Specifically, we repeat the experiments whose results are reported in Fig. 5a, but where strategy $\mathcal{S}'_h$ – the one used for the removal of examples – is BADGE. Results are shown in Fig. 6. Unlike Fig. 5a, there is no transition point, as it is always beneficial to remove examples selected by BADGE rather than random examples. This is because the added value of all points used for training diminishes after training is completed.

5 Summary and Discussion

It has been shown in previous work that different active learning strategies are suited for different budgets. In this paper, we present a hybrid integrated method, which combines competitive methods from the two different domains in order to achieve an active learning strategy that is suitable for **all** budgets. Our main contribution is twofold: First, we provide a theoretical analysis of the transition between low and high budgets, suggesting that this phenomenon may be universal. The main technical contributions, motivated by this analysis, involve: (i) the design of an effective test to distinguish between the domains in a given scenario, (ii) practical implementation. Finally, the performance of our method is shown to match or surpass other active learning strategies at all budgets.

REFERENCES

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Shankar Sasrty. A convex optimization framework for active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 209–216, 2013.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pp. 1183–1192. PMLR, 2017.

Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *International Conference on Machine Learning*. PMLR, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

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*, pp. 9729–9738, 2020.

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 *Twenty-Third International FLAIRS Conference*, 2010.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32:7026–7037, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Online*, 2009.

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

Rafid Mahmood, Sanja Fidler, and Marc T Law. Low budget active learning via wasserstein distance: An integer programming approach. *arXiv preprint arXiv:2106.02968*, 2021.

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.

Hiranmayi Ranganathan, Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. Deep active learning for image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3934–3938. IEEE, 2017.

Christopher Schröder and Andreas Niekler. A survey of active learning for text classification using deep neural networks. *arXiv preprint arXiv:2008.07267*, 2020.

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

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pp. 1308–1318. PMLR, 2020.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5972–5981, 2019.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pp. 268–285. Springer, 2020.

Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. *arXiv preprint arXiv:2205.11320*, 2022.

APPENDIX

A DERIVATION OF TRANSITION POINTS

Recall that the mean generalization error of mixed strategy q is:

$$E_{\mathcal{L}}(\mathbb{T}) = p \cdot E(qB) + (1-p) \cdot E(\alpha(1-q)B).$$

for differentiable function E. The mixture coefficient q, which obtains the minimal generalization error, must satisfy

$$\begin{aligned} 0 = \frac{\partial E_{\mathcal{L}}(\mathbb{T})}{\partial q} &= pBE'(qB) - (1-p)\alpha BE'(\alpha(1-q)B) \\ \implies \quad \frac{E^{'}(qB)}{E^{'}(\alpha(1-q)B)} &= \frac{\alpha(1-p)}{p} \end{aligned} \tag{9}$$

The transition points can now be defined as follows:

- B_{equiv} is obtained by solving (9) with $q = p$.
- B_{low} is obtained by solving (9) with $q = p + \frac{m(1-p)}{B}$.
- B_{high} is obtained by solving (9) with $q = p - \frac{m(1-p)}{B}$.

B HYPER-PARAMETERS

B.1 SUPERVISED TRAINING

When training on CIFAR-10 and CIFAR-100, we used a ResNet-18 trained over 50 epochs. We used an SGD optimizer, with 0.9 Nesterov momentum, 0.0003 weight decay, cosine learning rate scheduling with a base learning rate of 0.025, and batch size of 100 examples. We used random croppings and horizontal flips for augmentations. An example use of these parameters can be found at (Munjal et al., 2020).

When training ImageNet-50, we used the same hyper-parameters as CIFAR-10/100, only changing the base learning rate to 0.01 and the batch size to 50.

B.2 UNSUPERVISED REPRESENTATION LEARNING

CIFAR-10/100 We trained SimCLR using the code provided by (Van Gansbeke et al., 2020) for CIFAR-10 and CIFAR-100. Specifically, we used ResNet18 with an MLP projection layer to a 128 vector, trained for 500 epochs. All the training hyper-parameters were identical to those used by SCAN. After training, we used the 512 dimensional penultimate layer as the representation space. As in SCAN, we used an SGD optimizer with 0.9 momentum and an initial learning rate of 0.4 with a cosine scheduler. The batch size was 512 and a weight decay of 0.0001. The augmentations were random resized crops, random horizontal flips, color jittering, and random grayscaling. We refer to (Van Gansbeke et al., 2020) for additional details. We used the L2 normalized penultimate layer as embedding.

ImageNet-50 We extracted embedding from the official (ViT-S/16) DINO weights pre-trained on ImageNet. We used the L2 normalized penultimate layer as embedding.

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 HIGH BUDGET STRATEGIES

In Section. 4, we are required to use a high budget strategy $\mathcal{S}'_h$, which relies in its computation only on the unlabeled set $\mathbb{U}$. We use *reverse-TypiClust*, which is calculated similarly to *TypiClust*, only

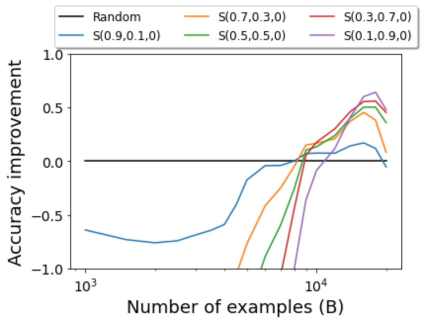


Figure 7: Accuracy gain by *reverse-TypiClust*, as compared to random query sampling.

using the most atypical example at each iteration instead of the most typical example. In Fig. 7, we plot the performance of such a strategy on CIFAR-10, as a function of budget B, similarly to the analysis done in Fig. 4.

We see that while *reverse-TypiClust* is not a competitive high-budget strategy, it still outperforms random sampling in the high-budget regime, making it a suitable AL strategy for this regime.

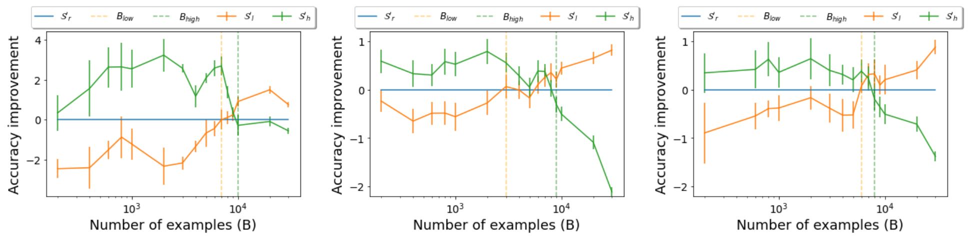
C.2 Other feature spaces

C.2.1 Removing data according to other feature spaces

In section 3.2, we propose an active learning method that determines the budget size by removing examples in a given feature space. The feature space used in section 3.2 was obtained by SimCLR, as these features proved beneficial to several low-budget active learning methods.

In this section, we check the dependency of MiSAL on the specific choice of feature space. In Fig. 8, we plot the strategy selection test as described in Alg. 1 in Section 3.1. The plotted results are trained on CIFAR-10. In order to generate the 3 subsets of labeled examples $data_l$, $data_h$ and $data_r$, we remove 5% of the labeled data (but not less than 1 datapoint per class). This test is done using 3 different feature spaces 1. MoCo (He et al., 2020), a transformer based approach. 2. SimCLR, as done in section 3.2. 3. SCAN (Van Gansbeke et al., 2020).

Similarly to the results reported in section 3.2, we can see that using any of the 3 feature spaces resulted in a similar result – MiSAL would behave similarly regardless of the choice of the underlying feature space.



(a) SimCLR (b) SCAN (c) MoCo

Figure 8: Similar to Fig. 5, but removing examples according to different feature spaces in CIFAR-10. Accuracy gain when using $\mathcal{S}'_l$ to select points for removal as compared to random selection (orange), or $\mathcal{S}'_h$ to select points for removal (green). Negative gain implies that the strategy is beneficial, and vice versa.

C.2.2 Other feature spaces in TypiClust

In Table 1 and Table 2, we plot the results of different AL strategies across different datasets and budgets. Low-budget strategies such as TypiClust and ProbCover require the choice of feature space to work properly. Following the original papers, we used the feature space given by SimCLR trained on the entire unlabeled pool $\mathbb{U}$.

To check whether the choice of the feature space affects the results of the low-budget performance, we trained TypiClust on the TinyImageNet dataset with various choices of feature spaces.

In Fig. 9, we plot 5 active learning iterations with an active set of $m = 1000$ of ResNet-50 trained on TinyImageNet. We considered 5 different feature spaces: 1. MoCo (He et al., 2020), a transformer based approach. 2. DINO (Caron et al., 2021), an SSL-based approach. 3. SimCLR, which was used in the original TypiClust paper. 4. SWAV an SSL-based approach. 5. A simple autoencoder on the pixel values (AE). We found that except for the AE, all methods perform similarly, suggesting that the choice of the representation space has little effect on the training of low-budget methods such as TypiClust.

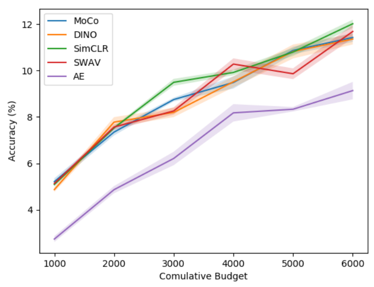


Figure 9: Comparing TypiClust with different representations on TinyImageNet for 5 AL iterations in the low budget regime. The active set size is $m = 1000$. The final test accuracy in each iteration is reported. The shaded area reflects standard error. All results are with respect to the 'random' representation, which is the pixel value of each image.