Active Learning on a Budget: Opposite Strategies Suit High and Low Budgets

Guy Hacohen[1 2 *] **Avihu Dekel**[1 *] **Daphna Weinshall**[1]

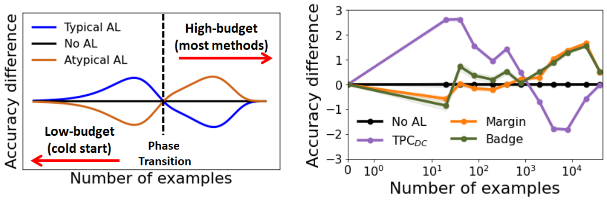
Abstract

Investigating active learning, we focus on the relation between the number of labeled examples (budget size), and suitable querying strategies. Our theoretical analysis shows a behavior reminiscent of phase transition: typical examples are best queried when the budget is low, while unrepresentative examples are best queried when the budget is large. Combined evidence shows that a similar phenomenon occurs in common classification models. Accordingly, we propose *TypiClust* – a deep active learning strategy suited for low budgets. In a comparative empirical investigation of supervised learning, using a variety of architectures and image datasets, *TypiClust* outperforms all other active learning strategies in the low-budget regime. Using *TypiClust* in the semi-supervised framework, performance gets an even more significant boost. In particular, state-of-the-art semi-supervised methods trained on CIFAR-10 with 10 labeled examples selected by *TypiClust*, reach 93.2% accuracy – an improvement of 39.4% over random selection. Code is available at https://github.com/avihu111/TypiClust.

1. Introduction

Recent years have witnessed the emergence of deep learning as the dominant force in advancing machine learning and its applications. But this development is data-hungry – to a large extent, deep learning owes its success to the growing availability of annotated data. This is problematic even in our era of *Big Data*, as the annotation of data remains costly.

Active Learning (AL) aims to alleviate this problem (see surveys in Settles, 2009; Schröder & Niekler, 2020). Given a large pool of unlabeled data, and possibly a small set of labeled examples, learning is done iteratively: the learner employs the labeled examples (if any), then queries an oracle by submitting examples to be annotated. This may be done repeatedly, often until a fixed budget is exhausted.



(a) Theoretical results (b) Empirical results

Figure 1. Visualization of phase transition in deep active learning, as revealed by plotting the difference in accuracy between different AL strategies to a random baseline as a function of budget size (the number of labeled examples). We see similar behavior both theoretically and empirically: when the budget is low, oversampling typical examples is more beneficial, whereas when the budget is high, oversampling atypical examples is more beneficial. (a) The behavior of an idealized model (see Section 2). (b) The behavior of *TypiClust*, contrasted with 2 basic uncertainty-based strategies, as seen in deep neural models trained on CIFAR-10 (see Section 4).

Many traditional active learning approaches are based on uncertainty sampling (e.g., Lewis & Gale, 1994; Ranganathan et al., 2017; Gissin & Shalev-Shwartz, 2019; Sinha et al., 2019). In uncertainty sampling, the learner queries examples about which it is least certain, presumably because such labels contain the most information about the problem. Another principle guiding deep AL approaches is diversity sampling (e.g., Hu et al., 2010; Elhamifar et al., 2013; Sener & Savarese, 2018; Shui et al., 2020). Here, queries are chosen to minimize the correlation between samples, in order to avoid redundancy in the annotations. Sensibly, diversity sampling is often combined with uncertainty sampling (see App. A for further discussion of related work).

At present, effective deep active learning strategies are known to require a large initial set of labeled examples to work properly (Yuan et al., 2020; Pourahmadi et al., 2021). We call this the *high budget regime*. In the *low budget regime*, where the initial labeled set is small or absent, it has been shown that random selection outperforms most deep AL strategies (see Attenberg & Provost, 2010; Mittal et al.,

[*] Equal contribution [1] School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel [2] Edmond & Lily Safra Center for Brain Sciences, The Hebrew University of Jerusalem, Jerusalem, Israel. Correspondence to: Guy Hacohen <guy.hacohen@mail.huji.ac.il>, Avihu Dekel <avihu.dekel@mail.huji.ac.il>, Daphna Weinshall <daphna@cs.huji.ac.il>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

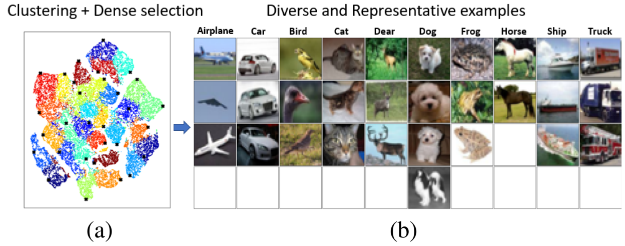
2019; Zhu et al., 2020; Siméoni et al., 2021; Chandra et al., 2021). This "cold start" phenomenon is often explained by the poor ability of neural models to capture uncertainty, which is more severe with a small budget of labels (Nguyen et al., 2015; Gal & Ghahramani, 2016). The low-budget scenario is relevant in many applications, especially those requiring an expert tagger whose time is expensive. If we want to expand deep learning to these domains, overcoming the cold start problem becomes an important challenge.

In this paper, we suggest that the low and high budgets regimes are qualitatively different, and require *opposite* querying strategies. Furthermore, we claim that the uncertainty principle is only suited for the high-budget regime, while the opposite strategy – the selection of the least ambivalent points – is suitable for the low-budget regime.

We begin, in Section 2, by establishing the theoretical foundations for this claim. We analyze a mixture model where two general learners, each limited to a distinct region of the input space, are independently learned. In this framework, we see a phase-transition-like phenomenon: in the low-budget regime, over-sampling the "easier" region, which can be learned from fewer examples, improves the outcome of learning. In the high-budget regime, over-sampling the alternative region is more beneficial. In other words, opposing querying strategies are suitable for the low-budget and high-budget regimes. This is illustrated in Fig. 1a.

We continue by identifying a set of sufficient conditions, which guarantee that two independent learners display this phase-transition-like phenomenon. We then give a formal argument, showing that linear classifiers satisfy these conditions. We further provide empirical evidence to the effect that neural models may also satisfy these conditions.

Previous art established that in the high-budget regime, it is beneficial to preferentially sample uncertain examples. The phase transition result predicts that in the low-budget regime, a strategy that preferentially samples the most certain examples is beneficial. However, estimating prediction certainty is difficult, and cannot be reliably accomplished in the low-budget regime with access to very few labeled examples. We therefore adopt an alternative approach, replacing the notion of *certainty* with the notion of *typicality* (loosely defined): a point is typical if it lies in a high-density region of the input space, irrespective of labels.



(a) (b)

Figure 2. Visualizing the selection of 30 examples from CIFAR-10 by *TypiClust*. (a) The data is first clustered into 30 clusters, and the densest region within every cluster is sampled. We show t-SNE dimensionality reduction of the feature space, colored by cluster assignment, where selected examples are marked by ×. (b) The selected images, organized column-wise by class. Note that the ensuing labeled set is approximately class-balanced, even though the queries are chosen without access to class labels.

In Section 3, guided by these observations, we propose Typical Clustering (*TypiClust*) – a strategy for active learning in the low-budget regime. *TypiClust* aims to pick a diverse set of typical examples, which are likely to be representative of the entire dataset. To this end, *TypiClust* employs self-supervised representation learning and then estimates each point's density in this representation. Diversity is obtained by clustering the dataset and sampling the densest point from each cluster (see Fig. 2).

In Section 4, we compare *TypiClust* to various AL strategies in the low-budget regime. *TypiClust* consistently improves generalization by a large margin, across different datasets and architectures, reaching state-of-the-art (SOTA) results in many problems. In agreement with Zhu et al. (2020), we also observe that the alternative AL strategies are not effective in this domain, and are even detrimental.

TypiClust is especially beneficial for semi-supervised learning. Although in the AL framework the learner has access to a big pool of unlabeled data by construction, most AL strategies do not exploit the unlabeled data for learning, beyond query selection. Recent studies report that the benefit of AL is marginal when incorporated into semi-supervised learning (Chan et al., 2021; Bengar et al., 2021), with little added value over the random selection of labels. Re-examining this observation, we note that semi-supervised learning is most beneficial in the low-budget regime, wherein the explored AL strategies are inherently not suitable. When incorporating *TypiClust*, which is designed for the low-budget regime, into semi-supervised learning, performance is indeed improved by a large margin.

Summary of Contribution

i. A novel theoretical model analyzing Active Learning (AL) as biased sampling strategies in a mixture model.

ii. Prediction of the cold start phenomenon in AL.

iii. Prediction that opposite strategies suit AL in the low-budget and high-budget regimes.

iv. Empirical support of these theoretical principles.

v. *TypiClust*, a novel strategy that significantly improves active learning in the low-budget regime.

vi. Large performance boost to SOTA semi-supervised methods by *TypiClust*.

2. Theoretical Analysis

Given a large pool of U unlabeled examples and a possibly small (or even empty) set of L labeled examples, an Active Learning (AL) method selects a small subset of B examples from U to submit as label queries to an oracle. We call the number of labeled examples known to the learner *budget*, whose size is $m = B + L$. In this section, we aim to study the optimal query selection strategy as it depends on m.

To this end, we analyze a mixture model of two general learners. In §2.1, the model is defined by first splitting the support of the data distribution into two distinct regions, R_1 and R_2, further assuming that each region is independently learned by its own general learner. R_1 and R_2 are distinguished by the property that if they are learned independently, R_1 is *easier to learn* than R_2 (as formalized in Def. 2 below). We then define the error score $E_{\mathcal{D}}(m)$ of this model, which measures the expected error of the model over all training samples of size m, as a function of m.

Within this framework, in §2.2 we derive a **threshold test** on the budget size m and $E_{\mathcal{D}}(m)$, which determines whether an optimal AL strategy should oversample R_1 or R_2. In §2.3 we obtain sufficient conditions on $E_{\mathcal{D}}(m)$, which guarantee phase transition as illustrated in Fig. 1a. In accordance, an optimal AL strategy will oversample R_1 if m is smaller than some threshold m_0, and oversample R_2 otherwise. We let the term *low budget regime* denote budgets with $m \leq m_0$, and *high budget regime* denote budgets with $m > m_0$. We may now conclude that for any learner model whose error score meets our sufficient conditions, opposite strategies suit the low and high budget regimes.

In §2.4 we analytically prove that the error score of a mixture of two linear classifiers satisfies the required conditions, while in App. D.2 we empirically show that this is the case also with deep neural networks.

2.1. Mixture Model Definition

We analyze a mixture of two learners model, where each learner is independently trained on a distinct part of the domain of the data distribution. Formally, let $\mathbf{X} = [\boldsymbol{x}, y]$ denote a single data point, where $\boldsymbol{x} \in \mathbb{R}^d$ denotes an input point and $y \in \mathbb{Y}$ its corresponding label or target value. For example, $\mathbb{Y}$ is $\mathbb{R}$ in regression problems, and $[k]$ in classification problems. Each point $\mathbf{X}$ is drawn from a distribution $\mathcal{D}$ with density $f_{\mathcal{D}}(\mathbf{X})$. We denote an i.i.d. sample of m points from $\mathcal{D}$ as $\mathbb{X}^m = \{\mathbf{X}_1, ..., \mathbf{X}_m\} \sim \mathcal{D}^m$.

Let $R_1, R_2 \subseteq \mathbb{R}^d \times \mathbb{Y}$ denote a partition of the domain of $f_{\mathcal{D}}$, where $R_1 \cup R_2 = \mathbb{R}^d \times \mathbb{Y}$ and $R_1 \cap R_2 = \emptyset$. Let $\mathcal{D}_1, \mathcal{D}_2$ denote the conditional distributions obtained when restricting $\mathcal{D}$ to regions R_1, R_2 respectively. Note that $\mathcal{D}$ can now be viewed as a mixture distribution, where points are sampled from $\mathcal{D}_1$ with probability $p = \int_{\mathbf{X} \in R_1} f_{\mathcal{D}}(\mathbf{X}) d\mathbf{X}$, and $\mathcal{D}_2$ with probability $(1-p)$. Let $m_i,\ i \in [2]$ denote the number of points in R_i when sampling m points from $\mathcal{D}$, and $\mathbb{X}^{m_i}$ denote the restriction of sample $\mathbb{X}^m$ to R_i. We denote the hypothesis of a learner when trained independently on $\mathbb{X}^{m_i}$ as $h(\mathbb{X}^{m_i})$.

Next, we define the error score of a learner, which is a function of m – the training set size. It measures the expected generalization error of the learner over all such training sets.

Definition 1 (Error score)**.** *Assume training sample $\mathbb{X}^m \sim \mathcal{D}^m$, with the corresponding learned hypothesis $h(\mathbb{X}^m)$ – a random variable whose distribution is denoted $\mathcal{D}_h$. Let $Er\left(h\left(\mathbb{X}^m\right)\right)$ denote the expected generalization error of this hypothesis. The expected error of the learner, over all training sets of size m, is given by*

$$E_{\mathcal{D}}(m) = \mathbb{E}_{\mathbb{X}^m \sim \mathcal{D}^m}\left[\mathbb{E}_{h(\mathbb{X}^m) \sim \mathcal{D}_h}\left[Er\left(h\left(\mathbb{X}^m\right)\right)\right]\right].$$

We adopt two common assumptions regarding $E_{\mathcal{D}}(m)$. (i) **Efficiency**: $E_{\mathcal{D}}(m)$ is strictly monotonically decreasing, namely, on average the learner benefits from additional examples. (ii) **Realizability**: $\lim\limits_{m \to \infty} E_{\mathcal{D}}(m) = 0$.

During training, we assume a mixture of independent learners in R_1, R_2, and a training set composed of m_1, m_2 examples from each region respectively. The error score of the mixture learner on $\mathcal{D}$ for $m = m_2 + m_1$ is:

$$E_{\mathcal{D}}(m) = p \cdot E_{\mathcal{D}_1}(m_1) + (1-p) \cdot E_{\mathcal{D}_2}(m_2). \quad (1)$$

As an important ingredient of the mixture model, we assume that one region requires fewer examples to be adequately learned, and call this region R_1. Essentially, we expect the error score to decrease faster at R_1, with $E_{\mathcal{D}_1}(m) < E_{\mathcal{D}_2}(m)\ \forall m$. Other than this difference, we expect the error score to be similar in R_1 and R_2.

Making this notion more precise, we define an order relation on partition R_1, R_2 as follows:

Definition 2 (Order $R_1 \prec R_2$)**.** *Let R_1, R_2 denote a partition of the domain of $f_{\mathcal{D}}$. Assume that the error score in R_1 and R_2 can be written as $E_{\mathcal{D}_1}(m) = E(m)$ and $E_{\mathcal{D}_2}(m) = E(\alpha m)$ for a single function $E(m)$ and $\alpha > 0$. We say that R_1 is easier to learn than R_2 and denote $R_1 \prec R_2$ if $\alpha < \frac{p}{1-p}$, where p is the probability of R_1.*

Note that $\alpha < 1$ if $p = 0.5$. We now assume that $R_1 \prec R_2$, and rewrite (1) as follows:

$$E_{\mathcal{D}}(m) = p \cdot E(m_1) + (1-p) \cdot E(\alpha m_2). \quad (2)$$

Finally, we extend $E(m)$ to domain $\mathbb{R}_{\geq 0}$ with the continuation of $E(m)$ denoted $E(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, which is in C^∞ and positive. **Efficiency** is extended to imply $E'(x) < 0$.

2.2. Deriving the Optimal Sampling Strategy

Considering the extended error score $E(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, we define a biased sampling strategy as follows:

$$m_1 = p \cdot m + \Delta, \qquad m_2 = (1-p) \cdot m - \Delta.$$

$\Delta = 0$ is essentially equivalent to random sampling from $\mathcal{D}$. $\Delta > 0$ implies that more training points are sampled from R_1 than R_2, and vice versa.

In Thm. 1 we show that to minimize the expected generalization error while considering a mixture model of two independent learners as defined above, choosing between the different sampling strategies can be done using a simple threshold test.

Theorem 1. *Given partition $R_1 \prec R_2$ and error score $E(x)$, let $p = Prob(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. The following threshold test decreases the error score for sample size m:*

$$\boxed{\frac{E'\left(pm\right)}{E'\left(\alpha\left(1-p\right)m\right)} \begin{cases} > \frac{\alpha(1-p)}{p} & \Longrightarrow \quad \text{over sample region } \mathrm{R}_1 \\ < \frac{\alpha(1-p)}{p} & \Longrightarrow \quad \text{over sample region } \mathrm{R}_2 \end{cases}}$$

Proof. Starting from (2), we obtain the test whereby the error score $E_{\mathcal{D}}(m)$ decreases when $\Delta > 0$

$$\begin{aligned} & p \cdot E\left(p \cdot m + \Delta\right) + (1-p) \cdot E\left(\alpha\left((1-p)\,m - \Delta\right)\right) \\ & \quad < p \cdot E\left(p \cdot m\right) + (1-p) \cdot E\left(\alpha\left(1-p\right)m\right) \\ \Longrightarrow\ & (1-p)\left[E\left(\alpha\left(1-p\right)m\right) - E\left(\alpha\left((1-p)\,m - \Delta\right)\right)\right] \\ & \quad > p\left[E\left(p \cdot m + \Delta\right) - E\left(p \cdot m\right)\right] \\ \Longrightarrow\ & \alpha\left(1-p\right)\frac{E\left(\alpha\left(1-p\right)m\right) - E\left(\alpha\left(1-p\right)m - \alpha\Delta\right)}{\alpha\Delta} \\ & \quad > p\frac{E\left(p \cdot m + \Delta\right) - E\left(p \cdot m\right)}{\Delta}. \end{aligned}$$

Since $E(x)$ is differentiable and strictly monotonically decreasing with $E' < 0$, in the limit of infinitesimal Δ

$$\frac{E'\left(pm\right)}{E'\left(\alpha\left(1-p\right)m\right)} > \frac{\alpha\left(1-p\right)}{p}.$$

The proof for $\Delta < 0$ is similar. □

Example: exponentially decreasing function. Assume $E\left(m\right) = e^{-m}$, and a mixture model with $p = 0.8, \alpha = 0.1$. We simulate the error in (2) when biasing the train sample with $\Delta = \pm 0.01$. Fig. 1a shows the differences between the error score of biased sampling (in favor of either R_1 in blue or R_2 in orange) and random sampling, as a function of the number of examples m. For small m it is beneficial to favorably bias region R_1, while for large m it is beneficial to favorably bias R_2. This is the behavior often seen in our empirical investigation, see Fig. 1b and Section 4.

2.3. Error Scores Analyzed

We now report sufficient conditions on $E(m)$, which guarantee the phase-transition-like behavior illustrated in Fig. 1a, starting with some formal definitions.

Given partition $R_1 \prec R_2$ and Error score $E(x)$, we say that $E(x)$ is undulating if it displays the following behavior: in the beginning, when the number of training examples is small, the generalization error decreases faster when oversampling region R_1. In the end, after seeing sufficiently many training examples, the generalization error decreases faster when over-sampling region R_2. Formally:

Definition 3 (Undulating)**.** *An error score $E(m)$ is undulating if there exist $z_1, z_2 \in \mathbb{R}$ such that $\frac{E'(pm)}{E'(\alpha(1-p)m)} > \frac{\alpha(1-p)}{p}$ $\forall m < z_1$, and $\frac{E'(pm)}{E'(\alpha(1-p)m)} < \frac{\alpha(1-p)}{p}$ $\forall m > z_2$.*

For undulating error scores, there could potentially be any number of transitions between the two conditions, switching the preference of R_1 to R_2, and vice versa. We extend the above definition to capture a case of particular interest, where this transition occurs only once, as follows:

Definition 4 (SP-undulating)**.** *An error score $E(m)$ is Single-Phase undulating if it is undulating, and $z_1 = z_2$.*

2.3.1. Undulating Error Scores

As motivated in Section 2.1, we define a proper error score as follows:

Definition 5 (Proper error score)**.** *$E(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a proper error score if it is a positive twice differentiable function, which is strictly monotonically decreasing ($E > 0$, $E' < 0$), $E(0) = c_0 \in \mathbb{R}_{>0}$, and where $\lim\limits_{x \to \infty} E(x) = 0$.*

Not all proper error scores will exhibit the phase undulating behavior. In Thm. 2 we state sufficient conditions that ensure this behavior (see proof in App. B.1).

Theorem 2 (Undulating error score: sufficient conditions)**.** *Given partition $R_1 \prec R_2$ and Error score $E(x)$, let $p = Prob(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. $E(x)$ is undulating if the following assumptions hold:*

(i) *$E(x)$ is a proper* error score *(see Def. 5).*
(ii) $\lim\limits_{x \to \infty} \frac{E'(x)}{E(x)}, \lim\limits_{x \to \infty} \frac{E(x)}{E(ax)}, \lim\limits_{x \to \infty} \frac{E'(x)}{E'(ax)}$ *exist* $\forall a \in (0, 1)$.
(iii) $-\log(E(x)) \in \omega\left(\log(x)\right)$.

Corollary 1 (Exponential error as a bound)**.** Error score *$E(x)$ is undulating if it satisfies assumptions (i) and (ii) of Thm. 2, and is bounded from above as follows*

$$E(x) \leq ke^{-\nu x} \qquad \forall x \in \mathbb{R}_{\geq 0}, \tag{3}$$

for some constants $\nu, k \in \mathbb{R}_{>0}$.

Proof. It can be readily verified that assumption (iii) of Thm. 2 follows from (3). □

2.3.2. SP-Undulating Error Scores

Thm. 3 extends the results of the previous section, by stating a set of sufficient conditions that ensure an SP-undulating error score. The proof can be found in App. B.2.

Theorem 3 (SP-undulating: sufficient conditions)**.** *Given partition $R_1 \prec R_2$ and Error score $E(x)$, let $p = Prob(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. $E(x)$ is SP-undulating if the following assumptions hold:*

1. *$E(x)$ is an undulating* error score.
2. *At least one of the following conditions holds:*
   - *(a) $\frac{-E''(x)\cdot x}{E'(x)}$ is monotonically increasing with x.*
   - *(b) $-E'(x)$ is strictly monotonically decreasing and log-concave.*

Corollary 2 (Exponential error is SP-undulating)**.** *Consider* error scores *of the form $E(x) = ke^{-\nu x}$ for constants $\nu, k \in \mathbb{R}_{>0}$. Such functions are SP-undulating.*

Cor. 2 shows that classifiers with an exponentially decreasing *error score* are SP-undulating, with a single transition from favoring R_1 to favoring R_2.

In practice, we cannot assume that the *error score* of commonly used learners is exponentially decreasing. However, frequently we can bound the *error score* from above by an exponentially decreasing function, as we demonstrate theoretically (see Section 2.4.1) and empirically (see Fig. 11 in App. D.2). In such cases, it follows from Cor. 1 that these functions are undulating.

2.4. Simple Classification Models

To make the analysis above more concrete, we analyze a mixture model of two linear classifiers in Section 2.4.1, as an example of an actual undulating model in common use. Additionally, we analyze the nearest neighbors classification model in Section 2.4.2, to shed light on the rationale behind the partition of the support to regions R_1 and R_2. This case analysis demonstrates specific circumstances that make it possible to learn from fewer examples in certain regions.

2.4.1. Mixture of Two Linear Classifiers

Consider a binary classification problem and assume a learner that delivers a mixture of two linear classifiers in $\mathbb{R}^d$. The two classifiers are obtained by independently minimizing the L_2 loss on points in R_1 and R_2 respectively.

(i) Bounding the error of each mixture component. We first derive a bound on the error separately for R_1 and R_2, as it depends on the sample size m_j for $j \in [2]$. Let $X \in \mathbb{R}^{d\times m_j}$ denote the matrix whose columns are the training vectors that lie in region R_j. Let $\boldsymbol{y} \in \{-1, 1\}^{m_j}$ denote a row labels vector, where 1 marks positive examples and -1 negative examples. The learner seeks a separating row vector $\hat{\boldsymbol{w}} \in \mathbb{R}^d$, where

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d} \|\boldsymbol{w}X - \boldsymbol{y}\|^2 \quad \Longrightarrow \quad \hat{\boldsymbol{w}} = \boldsymbol{y}X^\top(XX^\top)^{-1}.$$

In Thm. 4, we bound the error of a linear model by some exponential function of the number of training examples m_j. The proof and further details can be found in App. C.1.1.

Theorem 4 (Error bound on a linear classifier)**.** *Assume: (i) a bounded sample $\|\boldsymbol{x}_i\| \le \beta$, where $XX^\top$ is sufficiently far from singular so that its smallest eigenvalue is bounded from below by $\frac{1}{\Lambda}$; (ii) a realizable binary problem where the classes are separable by margin δ; (iii) full rank data covariance, where $\frac{1}{\lambda}$ denotes its smallest singular value. Then there exist some positive constants $k, \nu > 0$, such that $\forall m_j \in \mathbb{N}$ and every sample $\mathbb{X}^{m_j}$, the expected error of $\hat{\boldsymbol{w}}$ obtained using $\mathbb{X}^{m_j}$ is bounded by:*

$$\mathbb{E}_{\mathbf{X}\sim\mathcal{D}_j}[0-1 \text{ loss of } \hat{\boldsymbol{w}}] \le ke^{-\nu m_j}.$$

(ii) A mixture classifier. Assume a mixture of two linear classifiers, and let $E(m) = p\cdot E_{\mathcal{D}_1}(m_1) + (1-p)\cdot E_{\mathcal{D}_2}(m_2)$ denote its error score. The following theorem characterizes this function (the proof can be found in App. C.1.2):

Theorem 5 (Undulating error)**.** *Retain the assumptions of Thm. 4, and assume that $\forall a \in (0,1)$ the following limits exist $\lim_{m\to\infty}\frac{E'(m)}{E(m)}$, $\lim_{m\to\infty}\frac{E(m)}{E(am)}$, $\lim_{m\to\infty}\frac{E'(m)}{E'(am)}$. Then the error score of a mixture of two linear classifiers is undulating.*

In practice, the *error score* in this case is also SP-undulating, as demonstrated in Fig. 10 in App. D.2.

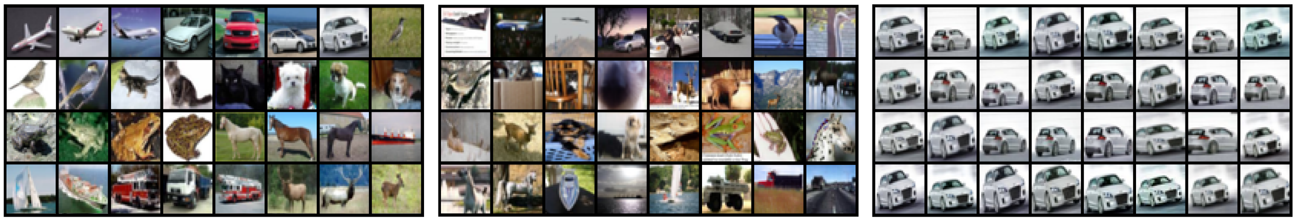
2.4.2. KNN Classifier and High-Density Regions

Our analysis in Section 2.3 shows that given some partition of the data into R_1 and R_2, where $R_1 \prec R_2$ (see Def. 2), then oversampling from R_1 is preferable in the low budget regime, while oversampling from R_2 is preferable in the high budget regime. To shed light on the nature of the assumed partition, we analyze below the discrete one-Nearest-Neighbor (1-NN) classification framework. Specifically, we show that **selecting R_1 as the set of the most probable points in the dataset** has the property that $R_1 \prec \{\Omega \setminus R_1\}$.

We further show in App. C.2 that in this framework, the selection of an initial pool of size m will benefit from the following heuristic:

- **Max density:** when selecting a point $\mathbf{X}_i$, maximize its density $f_{\mathcal{D}}(\mathbf{X}_i)$.
- **Diversity:** select points that are far apart, so that their corresponding sets of nearest neighbors do not overlap.

While 1-NN is a rather simplistic model, we propose to use the derived heuristics to guide deep active learning. In the rest of this paper, we show how these guiding principles benefit deep active learning in the low-budget regime.



(a) *TypiClust* – typical and diverse (b) A-typical and diverse (c) Typical and not diverse

Figure 3. Qualitative visualization of diversity and typicality in the low budget regime on CIFAR-10. (a) Diverse typical images chosen by *TypiClust*. (b) Picking the *least* typical example in each cluster. (c) Picking the most typical examples, without enforcing diversity.

3. Method: Low Budget Active Learning

In the low-budget regime, our theoretical analysis shows that it may be beneficial to bias the training sample in favor of certain regions in the data domain. It also establishes a connection between such regions and the principles of *max density* (or *typicality*) and *diversity*. Here, we incorporate these principles into a simple new active learning strategy called *TypiClust*, designed for the low-budget regime.

3.1. Framework and Definitions

Let L_0 denote an initial labeled set of examples, and U_0 denote an initial unlabeled pool. Active learning is done iteratively: at each iteration i, a set of B unlabeled examples is picked according to some strategy. These examples are annotated by an oracle, added to L_{i-1}, and removed from U_{i-1}. This process is repeated until the labels budget is exhausted, or some predefined termination conditions are satisfied. In the low-budget regime, the total number of labeled examples $|L_{i-1}| + B$ is assumed to be small. The case where $L_0 = \emptyset$ is called "initial pool selection".

To capture the principle of max density, we define the *Typicality* of an example by its density in some semantically meaningful feature space. Formally, we measure an example's *Typicality* by the inverse of the average Euclidean distance to its K nearest neighbors[1], namely:

$$Typicality(x) = \left(\frac{1}{K} \sum_{x_i \in K\text{-NN}(x)} ||x - x_i||_2 \right)^{-1}. \quad (4)$$

3.2. Proposed Strategy: Typical Clustering (*TypiClust*)

In the low-budget regime, an active learning strategy based on typical examples needs to overcome several obstacles: (a) Networks trained on only a few examples are prone to overfit, making measures of typicality noisy and unreliable. (b) Typical examples tend to be very similar, amplifying the need for diversity. The importance of typicality and diversity is visualized in Fig. 3.

To overcome these obstacles we propose a novel method, called *TypiClust*, which attempts to select typical examples while probing different regions of the data distribution. In our method, self-supervised representation learning is used to overcome (a), while clustering is used to overcome (b). *TypiClust* is therefore composed of three steps:

Step 1: Representation learning. Utilize the large unlabeled pool U_0 to learn a semantically meaningful feature space: first train a deep self-supervised task on $U_0 \cup L_0$, then use the penultimate layer of the resulting model as feature space. Such methods are commonly used for semantic feature extraction (Chen et al., 2020; Grill et al., 2020).

Step 2: Clustering for diversity. As *typicality* in (4) is evaluated by measuring distances to neighboring points, the most typical examples are usually close to each other, often resembling the same image (see Fig. 3c). To enforce diversity and thus better represent the underlying data distribution, we employ clustering. Specifically, at each AL iteration i, we partition the data into $|L_{i-1}| + B$ clusters. This choice guarantees that there are at least B clusters that do not intersect with the existing labeled examples. We refer to such clusters as *uncovered clusters*.

Step 3: Querying typical examples. We select the most typical examples from the B largest *uncovered clusters*. Selecting from *uncovered clusters* enforces diversity (also w.r.t L_{i-1}), while selecting the most typical example in each cluster favors the selection of representative examples.

As the steps above do not depend on any specific representation or clustering method, different variants of the *TypiClust* strategy can be constructed. Below we evaluate two variants, both of which outperform by a large margin the uncertainty-based strategies in the low-budget regime:

1. TPC_{DC}: Using a deep clustering algorithm both for the self-supervised and clustering tasks. In our experiments, we used SCAN (Van Gansbeke et al., 2020).
2. TPC_{RP}: Using representation learning followed by a clustering algorithm. We used DINO (Caron et al., 2021) for ImageNet, and SimCLR (Chen et al., 2020) for all other datasets, followed by K-means.

The pseudo-code of *TypiClust* for initial pool selection is

[1] We use $K = 20$, but other choices yield similar results.

given in Alg. 1 (see more details in App. F.1). Note that, unlike traditional active learning strategies, *TypiClust* relies on self-supervised representation learning, and therefore can be used for initial pool selection.

Algorithm 1 *TypiClust* initial pooling algorithm

```
**Input:** Unlabeled pool $U$, Budget $B$
**Output:** $B$ typical and diverse examples to query
Embedding $\leftarrow$ Representation_Learning($U$)
Clust $\leftarrow$ Clustering_algorithm(Embedding, $B$)
Queries $\leftarrow \emptyset$
**for all** $i = 1, ..., B$ **do**
    Add $\arg\max_{x \in \text{Clust}[i]} \{Typicality(x)\}$ to Queries
**end for**
**return** Queries
```

4. Empirical Study

We now report our empirical results. Section 4.1 describes the evaluation protocol, datasets, and baseline methods. Section 4.2 describes the actual experiments and results.

4.1. Methodology

We evaluate active learning separately in the following three frameworks. (i) **Fully supervised**: training a deep network solely on the labeled set, obtained by active queries. (ii) **Fully supervised with self-supervised embeddings**: training a linear classifier on the embedding obtained from a pre-trained self-supervised model. (iii) **Semi-supervised**: training a deep network on the labeled and unlabeled sets, using the competitive method FlexMatch (Zhang et al., 2021).

In (i) and (ii), we adopt the AL evaluation framework created by Munjal et al. (2020), which implements several AL methods including all baselines used here except *BADGE*. In (iii) we adopt the code and hyper-parameters provided by FlexMatch. As FlexMatch is computationally intensive, it was not evaluated on ImageNet, confining the study to datasets it was reported to handle. In all evaluated cases, *TypiClust* achieves large improvements (see App. F.2 for implementation details).

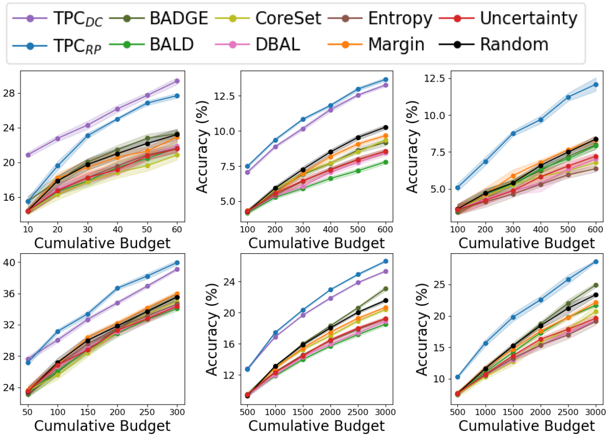
We compare *TypiClust* to the following baseline strategies for the selection of B points from U: (1) *Random* – uniformly. (2) *Uncertainty* – lowest max softmax output. (3) *Margin* – lowest margin between the two highest softmax outputs. (4) *Entropy* – highest entropy of softmax outputs. (5) *DBAL* (Gal et al., 2017). (6) *CoreSet* (Sener & Savarese, 2018). (7) *BALD* (Kirsch et al., 2019). (8) *BADGE* (Ash et al., 2020). All strategies are evaluated on the following image classification tasks: CIFAR-10/100 (Krizhevsky et al., 2009), TinyImageNet (Le & Yang, 2015) and ImageNet-50/100/200. The latter group includes subsets of ImageNet (Deng et al., 2009) containing 50/100/200 classes respectively, following Van Gansbeke et al. (2020).

4.2. Results: Low Budget Regime

The amount of labeled data that makes a budget "low" will vary between tasks. In the following experiments, unlike most earlier work, we focus on scenarios where on average, only 1-10 examples per class are labeled each round.

4.2.1. Fully Supervised Framework

Fig. 4 shows accuracy results for CIFAR-10/100 and ImageNet-100, using the labeled examples queried by different AL strategies. Denoting the number of classes by M, we show results with Budget $B = M$ or $B = 5M$ labeled examples and $L_0 = \emptyset$ (see App. G.1 for additional budgets).



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

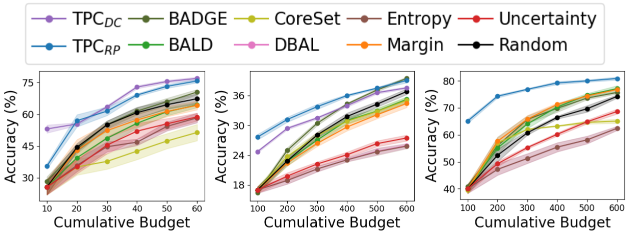
Figure 4. "Fully supervised" framework: comparing *TypiClust* with baseline AL strategies on CIFAR10, CIFAR100, and ImageNet-100 for 5 active learning iterations in the low budget regime. The budget B is equal to (top) the number of classes, or (bottom) 5 times the number of classes. The final average test accuracy in each iteration is reported, using 10 (CIFAR) and 3 (ImageNet) repetitions. The shaded area reflects standard error.

We see that in the low budget regime, both *TypiClust* variants outperform the baselines by a large margin. Specifically, all other baseline AL methods perform on par with random selection or worse, in accordance with Pourahmadi et al. (2021). In contrast, the typicality-based strategy achieves large accuracy gains. Noting that most of the baselines are possibly hampered by their use of random initial pool selection when $L_0 = \emptyset$, our ablation study in Section 4.3.1 demonstrates that this is not a decisive factor.

4.2.2. Fully Supervised with Self-Supervised Embedding

As self-supervised embeddings can be semantically meaningful, they are often used as features for a linear classifier.

Accordingly, in this framework, we use the extracted features from the representation learning step and train a linear classifier on the queried labeled set L_i. Unlike the fully supervised framework, here we use the unlabeled data while training the classifier, albeit in a basic manner. This framework outperforms the fully supervised framework, but still lags behind the semi-supervised framework. Once again, *TypiClust* outperforms all baselines by a large margin, as shown in Fig. 5 (see App. G.2 for additional datasets).



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

Figure 5. Similar to Fig. 4 in the "fully supervised with self-supervised embedding" framework.

4.2.3. Semi-Supervised Framework

In this framework, we evaluate *TypiClust* and different AL strategies by examining the performance of FlexMatch when trained on their respective queried examples. As semi-supervised methods often achieve competitive performance with only a few labeled examples, we focus on the extreme low-budget regime, where only $0.02\% \sim 1\%$ of the data is labeled. Note that semi-supervised algorithms typically assume a class-balanced labeled set, which is not feasible in active learning. To compare with this scenario which dominates the literature, we add a class-balanced random baseline for reference.

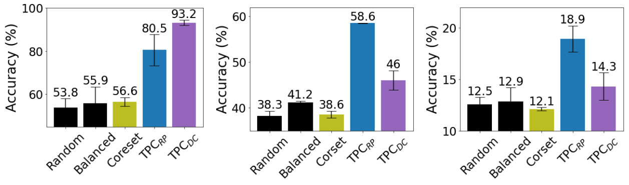
In Fig. 6, we compare the final performance of FlexMatch using the labeled sets provided by different AL strategies. We show results for a budget of 10 examples in CIFAR-10 (Fig. 6a), 300 examples in CIFAR-100 (Fig. 6b), and 1000 examples in TinyImageNet (Fig. 6c). We see that both *TypiClust* variants outperform random sampling, whether balanced or not, by a large margin. In contrast, other AL baselines do not improve the results of random sampling. Similar results using additional budgets, baselines, datasets, and semi-supervised algorithms, can be found in App. G.3.

4.3. Ablation Study

We now report the results of a set of ablation studies, checking the added value of each step in our suggested strategy.

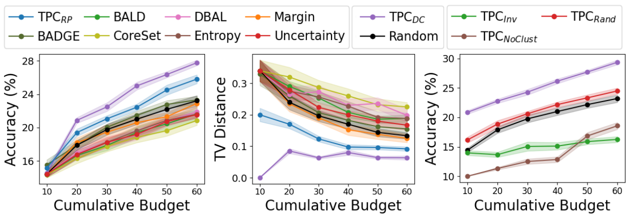
4.3.1. Random Initial Pool Selection

As *TypiClust* is based on self-supervised learning, both its variants are well suited for the case $L_0 = \emptyset$, and can actively query the initial selection of labeled examples. By contrast, the other AL baselines use random initial pool selection when $L_0 = \emptyset$. To isolate the effect of this difference, we conducted the same experiment as reported in Fig. 4a, giving *TypiClust* a random initial pool selection just like the other baselines. Results are reported in Fig. 7a, showing that *TypiClust* still outperforms all baselines. Importantly, this comparison reveals that non-random initial pool selection yields further generalization gains when combined with active learning. Additional results can be seen in Fig. 16.



(a) CIFAR-10 (b) CIFAR-100 (c) TinyImageNet

Figure 6. Comparison of AL strategies in a semi-supervised task. Each bar shows the mean test accuracy after 3 repetitions of FlexMatch trained on: (a) 10 examples from CIFAR-10, (b) 300 examples from CIFAR-100, (c) 1000 examples from TinyImageNet. Error bars show the standard error.



(a) (b) (c)

Figure 7. (a-b) Same experiment as in Fig. 4a top, but where: (a) *TypiClust* uses random initial set selection; (b) the Total Variation (TV) distance between the labeled set distribution and the ground truth class distribution is shown. (c) To isolate the added value of clustering for diversity and typical sample selection, we evaluate 3 additional selection heuristics on CIFAR-10 (see Section 4.3.3).

4.3.2. Comparing Class Distribution

With an extremely low budget, covering the support of the distribution comprehensively is challenging. To compare the success of the different AL strategies in this task, we measure the Total Variation (TV) distance between the labeled set class distribution and the ground truth class distribution for each strategy. Fig. 7b shows that the *TypiClust* variants achieve a significantly better (lower) score than the alternatives, resulting in queries with better class balance.

4.3.3. The Importance of Density and Diversity

TypiClust clusters the dataset and selects the most typical examples from every cluster. To assess the added value of clustering and typicality selection, we consider the following alternative selection criteria: (a) Select a random example from each cluster (TPC$_{Rand}$). (b) Select the most atypical example in every cluster (TPC$_{Inv}$). (c) Select typical samples greedily, without clustering (TPC$_{NoClust}$).

The results in Fig. 7c show that both clustering and high-density sampling are crucial for the success of *TypiClust*. The low performance of TPC$_{Rand}$ shows that representation learning and clustering alone cannot account for all the performance gain, while the low performance of TPC$_{NoClust}$ shows that typicality without diversity is not sufficient (see a visualization of these variants in Fig. 3).

4.3.4. Uncertainty Delivered by an Oracle

When trained on only a few labeled examples, neural networks tend to overfit, which may result in the unreliable estimation of uncertainty. This offers an alternative explanation to our results – uncertain examples may be a good choice in the low-budget regime as well, if only we could compute uncertainty accurately.

To test this hypothesis we first train an "oracle" network (see Lowell et al. (2018) and App. F.3) on the entire CIFAR-10 dataset and use its softmax margin to estimate uncertainty. This "oracle margin" is then used to choose the query examples. Subsequently, another network is trained similarly to the setup of Fig. 4a, adding in each iteration the examples with either the highest or lowest softmax response margin according to the oracle.

The results are shown in Fig. 8. We see that even a reliable measure of uncertainty leads to poor performance in the low-budget regime, even worse than the baseline uncertainty-based methods. This may be because these methods compute the uncertainty in an unreliable way, and thus behave more like the random selection strategy.

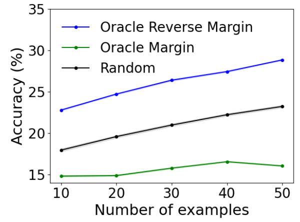


Figure 8. Certainty, as estimated by the margin of an oracle that knows all the labels, is used for AL. We plot the mean test accuracy of 100 models trained on CIFAR-10, $|L_0| = 10$, $B = 10$. STE is very small, as shown.

4.3.5. Imbalanced Data

Unsupervised representation learning methods often assume class-balanced datasets. As *TypiClust* is based on representation learning, it could potentially fail in imbalanced settings. We repeated our experiments on the class-imbalanced subset of CIFAR-10 proposed by Munjal et al. (2020). As before, we show that *TypiClust* outperforms other methods in the low-budget regime, and under-performs in the high-budget regime (see low budget results in Fig. 17 of App. G.1).

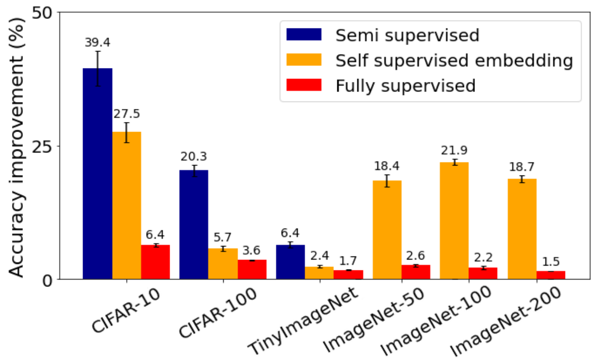


Figure 9. TypiClust achieves major accuracy gains as compared to the random selection baseline in the fully-supervised (3 repetitions on ImageNet, 10 otherwise), semi-supervised (3 reps) and self-supervised embedding (5 reps) frameworks. We use 10, 300, 1000, 50, 100, 200 examples in CIFAR-10, CIFAR-100, TinyImageNet, ImageNet-50, ImageNet-100 and ImageNet-200 respectively.

5. Summary and Discussion

We show, theoretically and empirically, that strategies for active learning in the high and low-budget regimes should be based on opposite principles. Initially, in the low-budget regime, the most typical examples, which the learner can learn most easily, are the most helpful to the learner. When reaching the high-budget regime, the best examples to query are those that the learner finds most confusing. This is the case both in the fully supervised and semi-supervised settings: we show that semi-supervised algorithms get a significant boost from seeing the labels of typical examples. Fig. 9 summarizes all our empirical results.

Our results are closely related to curriculum learning (Bengio et al., 2009; Hacohen & Weinshall, 2019; Weinshall & Amir, 2020), hard data mining, and self-paced learning (Kumar et al., 2010), all of which reflect the added value of typical ("easy") examples when there is little information about the task, as against atypical ("hard") examples which are more beneficial later on. Our results are also closely related to the study of the learning order of neural networks, which also characterize "easy" and "hard" examples Gissin & Shalev-Shwartz (2019); Hacohen et al. (2020); Shah et al. (2020); Hacohen & Weinshall (2021); Choshen et al. (2021).

The point of transition – what makes the budget "small" or "large", depends on the task and corresponding data distribution. In complex real-life problems, the low-budget regime may still contain a large number of examples, increasing the practicality of our method. Determining the range of training sizes with "low budget" characteristics is a challenging problem, which we leave for future work.

Acknowledgments

This work was supported by the Israeli Ministry of Science and Technology, and by the Gatsby Charitable Foundations.

References

Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., and Agarwal, A. 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.

Attenberg, J. and Provost, F. Why label when you can search? alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 423–432, 2010.

Bengar, J. Z., van de Weijer, J., Twardowski, B., and Raducanu, B. Reducing label effort: Self-supervised meets active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1631–1639, 2021.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

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

Chandra, A. L., Desai, S. V., Devaguptapu, C., and Balasubramanian, V. N. On initial pools for deep active learning. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pp. 14–32. PMLR, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Choshen, L., Hacohen, G., Weinshall, D., and Abend, O. The grammar-learning trajectories of neural language models. *arXiv preprint arXiv:2109.06096*, 2021.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

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

Gao, M., Zhang, Z., Yu, G., Arık, S. Ö., Davis, L. S., and Pfister, T. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*, pp. 510–526. Springer, 2020.

Geifman, Y. and El-Yaniv, R. Deep active learning over the long tail. *arXiv preprint arXiv:1711.00941*, 2017.

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

Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. Á., Guo, Z., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent - A new approach to self-supervised learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Hacohen, G. and Weinshall, D. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pp. 2535–2544. PMLR, 2019.

Hacohen, G. and Weinshall, D. Principal components bias in deep neural networks. *arXiv preprint arXiv:2105.05553*, 2021.

Hacohen, G., Choshen, L., and Weinshall, D. Let's agree to agree: Neural networks share classification order on real datasets. In *International Conference on Machine Learning*, pp. 3950–3960. PMLR, 2020.

He, T., Jin, X., Ding, G., Yi, L., and Yan, C. Towards better uncertainty sampling: Active learning with multiple views for deep convolutional neural network. In *2019*

IEEE International Conference on Multimedia and Expo (ICME), pp. 1360–1365. IEEE, 2019.

Hong, S., Ha, H., Kim, J., and Choi, M.-K. Deep active learning with augmentation-based consistency estimation. *arXiv preprint arXiv:2011.02666*, 2020.

Hu, R., Mac Namee, B., and Delany, S. J. Off to a good start: Using clustering to select the initial training set in active learning. In *Twenty-Third International FLAIRS Conference*, 2010.

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

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

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *NIPS*, volume 1, pp. 2, 2010.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Lerner, B., Shiran, G., and Weinshall, D. Boosting the performance of semi-supervised learning with unsupervised clustering. *arXiv preprint arXiv:2012.00504*, 2020.

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

Lowell, D., Lipton, Z. C., and Wallace, B. C. Practical obstacles to deploying active learning. *arXiv preprint arXiv:1807.04801*, 2018.

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

Mittal, S., Tatarchenko, M., Çiçek, Ö., and Brox, T. Parting with illusions about deep active learning. *arXiv preprint arXiv:1912.05361*, 2019.

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

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Park, S., Han, S., Kim, S., Kim, D., Park, S., Hong, S., and Cha, M. Improving unsupervised image clustering with robust learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12278–12287, 2021.

Pourahmadi, K., Nooralinejad, P., and Pirsiavash, H. A simple baseline for low-budget active learning. *arXiv preprint arXiv:2110.12033*, 2021.

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

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

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

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

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33: 9573–9585, 2020.

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

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

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

Tropp, J. A. An introduction to matrix concentration inequalities. *Found. Trends Mach. Learn.*, 8(1-2):1–230, 2015. doi: 10.1561/2200000048.

Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., and Van Gool, L. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pp. 268–285. Springer, 2020.

Wang, Z., Du, B., Zhang, L., and Zhang, L. A batch-mode active learning framework by querying discriminative and representative samples for hyperspectral image classification. *Neurocomputing*, 179:88–100, 2016.

Weinshall, D. and Amir, D. Theory of curriculum learning, with convex loss functions. *Journal of Machine Learning Research*, 21(222):1–19, 2020.

Yang, Y., Ma, Z., Nie, F., Chang, X., and Hauptmann, A. G. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015.

Yehuda, O., Dekel, A., Hacohen, G., and Weinshall, D. Active learning through a covering lens. *arXiv preprint arXiv:2205.11320*, 2022.

Yin, C., Qian, B., Cao, S., Li, X., Wei, J., Zheng, Q., and Davidson, I. Deep similarity-based batch mode active learning with exploration-exploitation. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 575–584. IEEE, 2017.

Yuan, M., Lin, H., and Boyd-Graber, J. L. Cold-start active learning through self-supervised language modeling. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 7935–7948. Association for Computational Linguistics, 2020.

Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., and Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *CoRR*, abs/2110.08263, 2021.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhdanov, F. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*, 2019.

Zhu, Y., Lin, J., He, S., Wang, B., Guan, Z., Liu, H., and Cai, D. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Trans. Knowl. Data Eng.*, 32 (4):631–644, 2020. doi: 10.1109/TKDE.2019.2891530.

.

Appendix

A. Related Work

A.1. Diversity Sampling

Deep AL strategies often enforce diversity on the queried batch. The motivation is to avoid redundancy in the annotations, and to represent all parts of the training distribution. Sener & Savarese (2018) introduced the coreset approach, querying examples that cover the training distribution in a greedy manner. Many other AL algorithms incorporate diversity as part of their sampling strategy, including: Hu et al. (2010); Elhamifar et al. (2013); Yang et al. (2015); Wang et al. (2016); Yin et al. (2017); Zhdanov (2019); He et al. (2019); Kirsch et al. (2019); Ash et al. (2020); Shui et al. (2020). Notably, as deep active learning is practical only in batch settings, the importance of diversity is amplified (Geifman & El-Yaniv, 2017; Sener & Savarese, 2018).

Diversity sampling is orthogonal to uncertainty sampling, and can be added to almost any strategy. As opposed to previous works, our strategy aims to query a diverse set of characteristic examples, while other strategies aim to achieve diverse sets of uncharacteristic examples.

A.2. AL Strategies for Low Budgets

Recently, AL in the low-budget regime received increased attention. Strategies designed to address this regime usually employ self-supervised or semi-supervised methods using the unlabeled pool (Gao et al., 2020; Hong et al., 2020; Mahmood et al., 2021; Yehuda et al., 2022). The embedding of such methods is often utilized by methods that estimate uncertainty, as it gives an informative distance measure (Zhang et al., 2018).

In particular, Yuan et al. (2020) aims to solve the cold-start problem by using the embedding of a pre-trained model on some unsupervised task. Their experiments use pre-trained language models, with a strategy that decreases the dependency on high budgets but is still faithful to uncertainty sampling. Mahmood et al. (2021) suggested querying a diverse set of examples with minimal Wasserstein distance from the unlabeled pool. They report a significant performance boost in the low-budget regime. Unlike our work, they conduct experiments only in a fully supervised with self-supervised embedding settings, related to, but somewhat different from, the one described in Section 4.2.2.

B. Mixture Model Lemmas and Proofs

B.1. Undulating Error Score: Sufficient Conditions

Below we provide the proof for Thm. 2, which is stated in Section 2.3, and which lists sufficient conditions for error scores to be undulating (see Def. 3). We start with a few lemmas that will be used in this proof.

Lemma 1. *Let $f:\mathbb{R}_{\geq 0}\to\mathbb{R}$ denote a differentiable function with $f(0)\neq 0$. Then*

$$\lim_{x\to 0^+}\frac{f\left(x\right)}{f\left(ax\right)}=1\quad\forall a\in(0,1).$$

Proof. Omitted. □

Lemma 2. *Let $F:\mathbb{R}_{\geq 0}\to\mathbb{R}_{>0}$ and $f=F'$ denote a positive differentiable strictly monotonically decreasing function ($F>0, f<0$). Assume that $\lim_{x\to\infty}F(x)=0$, and that the limits $\lim_{x\to\infty}\frac{F(x)}{F(ax)}, \lim_{x\to\infty}\frac{f(x)}{f(ax)}$ exist $\forall a\in(0,1)$. Denote $g(x)=-\ln\left(F(x)\right)$. If*

$$g'(x)\in\omega\left(\frac{1}{x}\right),$$

then:

$$\lim_{x\to\infty}\frac{f(x)}{f\left(ax\right)}=0\quad\forall a\in(0,1).$$

Proof. We can write $F(x)=e^{-g(x)}$. It follows from the mean value theorem that $\exists t\ ax<t<x$ such that

$$\begin{aligned}\frac{F(x)}{F(ax)}&=e^{-(g(x)-g(ax))}\\&=e^{-(g'(t)x(1-a))}\\&=e^{-(g'(t)t\cdot\frac{x}{t}\cdot(1-a))}.\end{aligned}$$

Since $g'(x)\in\omega\left(\frac{1}{x}\right)$ we get

$$\lim_{t\to\infty}t\cdot g'(t)=\infty.$$

As $(1-a)<\frac{x}{t}(1-a)<\frac{(1-a)}{a}$, it follows that

$$\lim_{x\to\infty}\frac{F(x)}{F(ax)}=0.$$

From the assumption that the limits exist, and since $\lim_{x\to\infty}F(x)=\lim_{x\to\infty}F(ax)=0$, we can use L'Hôpital's rule and get

$$\lim_{x\to\infty}\frac{F(x)}{F(ax)}=\lim_{x\to\infty}\frac{f(x)}{af(ax)}=\frac{1}{a}\lim_{x\to\infty}\frac{f(x)}{f(ax)}=0.$$

□

Lemma 3. *Let $F:\mathbb{R}_{\geq 0}\to\mathbb{R}_{>0}$ denote a positive differentiable function ($F>0$). Denote $g(x)=-\ln(F(x))$. Assume that $\lim_{x\to\infty}g'(x)$ exists, and*

$$g(x)\in\omega\left(\log(x)\right),$$

then

$$g'(x)\in\omega\left(\frac{1}{x}\right).$$

Proof. $g(x) \in \omega(\log(x))$ implies that

$$\lim_{x\to\infty} \frac{g(x)}{\ln(x)} = \infty, \quad \lim_{x\to\infty} g(x) = \infty.$$

We can now use L'Hôpital's rule and get

$$\infty = \lim_{x\to\infty} \frac{g(x)}{\ln(x)} = \lim_{x\to\infty} \frac{g'(x)}{\frac{1}{x}} = \lim_{x\to\infty} xg'(x).$$

□

Theorem 2. *Given partition $R_1 \prec R_2$ and Error score $E(x)$, let $p = Prob(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. $E(x)$ is undulating if the following assumptions hold:*

(i) *$E(x)$ is a proper* error score *(see Def. 5).*
(ii) $\lim\limits_{x\to\infty} \frac{E'(x)}{E(x)}, \lim\limits_{x\to\infty} \frac{E(x)}{E(ax)}, \lim\limits_{x\to\infty} \frac{E'(x)}{E'(ax)}$ *exist* $\forall a \in (0, 1)$.
(iii) $-\log(E(x)) \in \omega(\log(x))$.

Proof. We define $f(x) = -E'(px), a = \frac{\alpha(1-p)}{p} < 1$. From assumption (i) and using Lemma 1, we get

$$\lim_{x\to 0^+} \frac{E'(px)}{E'(\alpha(1-p)x)} = 1.$$

Therefore there exists some $z_1 \in \mathbb{R}_{\geq 0}$ such that $\forall x < z_1$

$$\frac{E'(px)}{E'(\alpha(1-p)x)} > \frac{\alpha(1-p)}{p}.$$

From assumptions (i)-(iii) and using Lemmas 2-3, we get

$$\lim_{x\to\infty} \frac{E'(x)}{E'(ax)} = \lim_{x\to\infty} \frac{E'(px)}{E'(\alpha(1-p)x)} = 0,$$

and therefore there is some $z_2 \in \mathbb{R}_{\geq 0}$ such that $\forall x > z_2$

$$\frac{E'(px)}{E'(\alpha(1-p)x)} < \frac{\alpha(1-p)}{p}.$$

From Def. 3 we get that $E(x)$ is undulating. □

B.2. SP-undulating Error Score: Sufficient Conditions

We provide next the proof for Thm. 3, which is stated in Section 2.3.2, and which lists sufficient conditions for error scores to be SP-undulating (see Def. 3), extending Thm. 2. Once again, we start with a few lemmas.

Lemma 4. *Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ denote a positive differentiable function ($f > 0$). Let $0 < a < 1$ denote some constant. If*

$$h(x) = \frac{-f'(x)x}{f(x)}$$

is strictly monotonically increasing, then

$$g(x) = \frac{f(x)}{f(ax)}$$

is strictly monotonically decreasing.

Proof. $g(x)$ is monotonically decreasing iff $g'(x) < 0$, where

$$g'(x) = \frac{f'(x)f(ax) - af(x)f'(ax)}{f(ax)^2}.$$

This condition translates to

$$f'(x)f(ax) - af(x)f'(ax) < 0.$$

As by assumption $h(x)$ is monotonically increasing and $h(ax) < h(x)$, we get that $\forall x > 0$

$$\begin{aligned} & \frac{-f'(ax)ax}{f(ax)} < \frac{-f'(x)x}{f(x)} \\ \implies & \frac{-f'(ax)a}{f(ax)} < \frac{-f'(x)}{f(x)} \\ \implies & -f'(ax)f(x)a < -f'(x)f(ax). \end{aligned}$$

□

Lemma 5. *Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ denote a positive differentiable log-concave function which is strictly monotonically decreasing ($f > 0, f' < 0, (\log(f))'' \leq 0$). Then the following function is strictly monotonically increasing*

$$h(x) = \frac{-f'(x)x}{f(x)}.$$

Proof. $h(x)$ is strictly monotonically increasing iff $h'(x) > 0$, which holds iff

$$\begin{aligned} h'(x) = \frac{xf'(x)^2 - xf(x)f''(x) - f(x)f'(x)}{f(x)^2} > 0 \\ \implies \quad x\left[f'(x)^2 - f(x)f''(x)\right] - f(x)f'(x) > 0. \end{aligned}$$

Recall that $x > 0$ and $-f(x)f'(x) > 0 \ \forall x \in \mathbb{R}_{\geq 0}$. Since f is log-concave, we also have that $f'(x)^2 - f(x)f''(x) \geq 0$, which concludes the proof. □

Theorem 3. *Given partition $R_1 \prec R_2$ and Error score $E(x)$, let $p = Prob(R_1)$ and $0 < \alpha < \frac{p}{1-p}$. $E(x)$ is SP-undulating if the following assumptions hold:*

1. $E(x)$ is an undulating proper *error score*.
2. At least one of the following conditions holds:
   (a) $\frac{-E''(x)\cdot x}{E'(x)}$ is monotonically increasing with x.
   (b) $-E'(x)$ is strictly monotonic decreasing and log-concave.

Proof. Define the following positive continuous function

$$H(x) = \frac{E'(px)}{E'(\alpha(1-p)x)}.$$

Let $f(x) = -E'(x)$, $a = \frac{a(1-p)}{p} < 1$. Note that assumption 2a follows from assumption 2b and Lemma 5. Assumption 2 therefore implies that $\frac{-f'(x)x}{f(x)}$ is strictly monotonically increasing, and by Lemma 4 we can conclude that $H(x)$ is monotonically decreasing. Together with assumption 1, $H\left(x\right) = \frac{\alpha(1-p)}{p}$ at a single point, and we may therefore conclude that $E(x)$ is SP-undulating. □

Corollary 3. *If p – the probability of region R_1 – is sufficiently small so that $p < \frac{\alpha}{1+\alpha}$, then the conclusions are reversed: it is beneficial to initially over-sample R_2, and vice versa.*

C. Error Function of Simple Mixture Models

C.1. Mixture of Two Linear Classifiers

We next prove Thm. 4 and Thm. 5, which are stated in Section 2.4.1. Thm. 4 provides a bound on the error score of a single linear classifier, showing that under mild conditions, this score is bounded by an exponentially decreasing function in the number of training examples m. Thm. 5 states conditions under which the error score of a mixture of two linear models $E(m)$ is undulating, and presents the phase-transition behavior.

C.1.1. Bounding the Error of Each Mixture Component

Henceforth we use the notations of Section 2.4.1, where for clarity, m_j is replaced by m while we are discussing the bound on a single component $j \in [2]$. Let $\boldsymbol{x}_i$ denote the i-th data point and i-th column of X. Let μ_1 and μ_2 denote the respective means of the two classes, and $\mu = \mu_1 - \mu_2$ denote the vector difference between the means.

Assuming that the data is normalized to 0 mean, the maximum likelihood estimators for the covariance matrix of the distribution Σ and class means, denoted $\hat{\Sigma}$ and $\hat{\mu}_1, \hat{\mu}_2$ respectively, are the following

$$\hat{\Sigma} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^\top = \frac{1}{m} X X^\top \tag{5}$$

$$\hat{\mu}_j = \frac{1}{m_j}\sum_{\boldsymbol{x}_i \in C_j} \boldsymbol{x}_i \implies \boldsymbol{y} X^\top = m_1\hat{\mu}_1 - m_2\hat{\mu}_2, \tag{6}$$

where C_j denotes the set of points in class $j \in [2]$. Thus, the ML linear separator can be written as

$$\begin{aligned}\hat{\boldsymbol{w}} &= \boldsymbol{y} X^\top (X X^\top)^{-1} \\ &= [m_1\hat{\mu}_1 - m_2\hat{\mu}_2]\ (m\hat{\Sigma})^{-1} = \hat{\mu}\hat{\Sigma}^{-1},\end{aligned}$$

where $\hat{\mu} = \frac{1}{m}\boldsymbol{y}X^\top$. Note that $\hat{\mu}$ is the sample mean of vectors $\{y_i \boldsymbol{x}_i\}_{i=1}^m$, and $\hat{\Sigma}$ is the sample covariance of $\{\boldsymbol{x}_i\}_{i=1}^m$.

When $d > m$ (fewer training points than the input space dimension), $\hat{\Sigma}$ is rank deficient and therefore $\hat{\Sigma}^{-1}$ is not defined. Moreover, the solution is not unique. Nevertheless, it can be shown that the minimal norm solution is the Penrose pseudo-inverse $\hat{\Sigma}^{+}$, where

$$\hat{\Sigma} = UDU^\top \implies \hat{\Sigma}^{+} = UD^{+}U^\top,$$

and using the notations

$$\begin{aligned} D &= \mathrm{diag}(d_1, \ldots, d_m, 0, \ldots, 0) \\ D^{+} &= \mathrm{diag}(d_1^{-1}, \ldots, d_m^{-1}, 0, \ldots, 0).\end{aligned}$$

Ignoring the question of uniqueness, estimating $\boldsymbol{w}$ is therefore reduced to evaluating the estimators in (5) and (6). These ML estimators have the following known upper bounds on their error:

1. Bounding $\hat{\Sigma}$: from known results on covariance estimation (Tropp, 2015), using Bernstein matrix inequality

   $$P(\|\hat{\Sigma} - \Sigma\|_{op} \geq t) \leq 2de^{-\gamma m t^2}. \tag{7}$$

   Constant γ does not depend on m; it is determined by the assumed bound on the L_2 norm of vectors $\boldsymbol{x}_i$, and the norm of the true covariance matrix Σ.
2. Bounding $\hat{\mu}$: starting from Hoeffding's inequality in one dimension, we have that $P(\left|\hat{\mu}^k - \mu^k\right| \geq t) \leq 2e^{\frac{-2mt^2}{4\beta^2}}$ $\forall k \in [d]$, where we assume a bounded distribution $\|\boldsymbol{x}\| \leq \beta$. Thus

   $$\begin{aligned} P(\|\hat{\mu} - \mu\| \geq t) &= P(\|\hat{\mu} - \mu\|^2 \geq t^2) \\ &= P(\sum_{k=1}^{d} \left|\hat{\mu}^k - \mu^k\right|^2 \geq t^2) \\ &\leq \sum_{k=1}^{d} P(\left|\hat{\mu}^k - \mu^k\right|^2 \geq \frac{t^2}{d}) \qquad (8) \\ &= \sum_{k=1}^{d} P(\left|\hat{\mu}^k - \mu^k\right| \geq \frac{t}{\sqrt{d}}) \\ &\leq 2de^{-2m\frac{t^2}{4\beta^2 d}} \end{aligned}$$

The first inequality follows from the union-bound inequality.

Lemma 6.

$$\hat{\mu}^\top[\Sigma^{-1} - \hat{\Sigma}^{+}]\boldsymbol{x} = \hat{\mu}^\top[\hat{\Sigma}^{+}(\hat{\Sigma} - \Sigma)\Sigma^{-1}]\boldsymbol{x}. \tag{9}$$

Proof. Because $\hat{\Sigma} \in \mathbb{R}^{d\times d}$ is of rank m, $Q = \hat{\Sigma}^{+}\hat{\Sigma} = U\mathrm{diag}(1, \ldots, 1, 0, \ldots, 0)U^\top$ is a projection matrix of rank m, projecting vectors to the subspace spanned by the training set $\{\boldsymbol{x}_i\}_{i=1}^m$. Thus $Q\hat{\mu} = \hat{\mu}$. Additionally, by definition, $\hat{\Sigma}^{+}\hat{\Sigma}\hat{\Sigma}^{+} = \hat{\Sigma}^{+}$ and Q is symmetric. It follows that

$$\hat{\mu}^\top(\Sigma^{-1} - \hat{\Sigma}^{+})\boldsymbol{x} = \hat{\mu}^\top(Q\Sigma^{-1} - \hat{\Sigma}^{+})\boldsymbol{x},$$

while

$$\hat{\Sigma}^{+}(\hat{\Sigma}-\Sigma)\Sigma^{-1} = Q\Sigma^{-1} - \hat{\Sigma}^{+}.$$

Together, we get (9). □

Theorem 4. *Assume: (i) a bounded sample $\|\boldsymbol{x}_i\| \leq \beta$, where $XX^\top$ is sufficiently far from singular so that its smallest eigenvalue is bounded from below by $\frac{1}{\Lambda}$; (ii) a realizable binary problem where the classes are separable by margin δ; (iii) full rank data covariance, where $\frac{1}{\lambda}$ denotes its smallest singular value. Then there exist some positive constants $k, \nu > 0$, such that $\forall m_j \in \mathbb{N}$ and every sample $\mathbb{X}^{m_j}$, the expected error of $\hat{\boldsymbol{w}}$ obtained using $\mathbb{X}^{m_j}$ is bounded by:*

$$F(m_j) = \mathbb{E}_{\mathbf{X}\sim\mathcal{D}_j}[0-1 \text{ loss of } \hat{\boldsymbol{w}}] \leq ke^{-\nu m_j}$$

Proof. It follows from our assumptions that $\|\hat{\Sigma}^{+}\|_{op} \leq \Lambda$. An error will occur at $\boldsymbol{x} \in C_1$ if $\boldsymbol{w}_{opt}\boldsymbol{x} \geq \delta$ and $\hat{\boldsymbol{w}}\boldsymbol{x} < 0$, and vice versa for $\boldsymbol{x} \in C_2$. In either case, the difference between the predictions of $\boldsymbol{w}_{opt}$ and $\hat{\boldsymbol{w}}$ deviate by more than δ. Thus

$$\begin{aligned} F(m) &= \mathbb{E}_{\mathbf{X}\sim\mathcal{D}}[0-1 \text{ loss}] = P(\text{error}) \\ &\leq P\left(\left|\mu^\top\Sigma^{-1}\boldsymbol{x} - \hat{\mu}^\top\hat{\Sigma}^{+}\boldsymbol{x}\right| > \delta\right). \end{aligned}$$

Invoking the triangular inequality

$$\begin{aligned} \Delta &= \left|\mu^\top\Sigma^{-1}\boldsymbol{x} - \hat{\mu}^\top\hat{\Sigma}^{+}\boldsymbol{x}\right| \\ &= \left|(\mu-\hat{\mu})^\top\Sigma^{-1}\boldsymbol{x} + \hat{\mu}^\top(\Sigma^{-1}-\hat{\Sigma}^{+})\boldsymbol{x}\right| \qquad (10) \\ &\leq \left|(\mu-\hat{\mu})^\top\Sigma^{-1}\boldsymbol{x}\right| + \left|\hat{\mu}^\top(\Sigma^{-1}-\hat{\Sigma}^{+})\boldsymbol{x}\right|. \end{aligned}$$

We now use Lemma 6 in order to shift $(\Sigma^{-1} - \hat{\Sigma}^{+})$ to $(\Sigma - \hat{\Sigma})$. Specifically, we insert (9) into (10) to get

$$\begin{aligned} \Delta &\leq \|\mu-\hat{\mu}\| \left\|\Sigma^{-1}\right\|_{op} \|\boldsymbol{x}\| \\ &\quad + \|\hat{\mu}\| \left\|\hat{\Sigma}^{+}\right\|_{op} \left\|\hat{\Sigma}-\Sigma\right\|_{op} \left\|\Sigma^{-1}\right\|_{op} \|\boldsymbol{x}\| \\ &\leq \|\mu-\hat{\mu}\| \lambda\beta + \beta\Lambda \left\|\hat{\Sigma}-\Sigma\right\|_{op} \lambda\beta. \end{aligned}$$

It follows that

$$\begin{aligned} &P\Big(\left|\mu^\top\Sigma^{-1}\boldsymbol{x} - \hat{\mu}^\top\hat{\Sigma}^{+}\boldsymbol{x}\right| > \delta\Big) \\ &\leq P(\|\mu-\hat{\mu}\| \lambda\beta + \left\|\hat{\Sigma}-\Sigma\right\|_{op} \lambda\Lambda\beta^2 > \delta) \\ &\leq P(\|\mu-\hat{\mu}\| \lambda\beta > \frac{\delta}{2}) + P(\left\|\hat{\Sigma}-\Sigma\right\|_{op} \lambda\Lambda\beta^2 > \frac{\delta}{2}) \\ &\leq 2de^{-2m\left(\frac{\delta}{2\lambda\beta}\right)^2 \frac{1}{4\beta^2 d}} + 2de^{-am\left(\frac{\delta}{2\lambda\Lambda\beta^2}\right)^2} \\ &\leq 4de^{-km\delta^2}. \end{aligned}$$

The second transition follows from the union-bound inequality, and the third from (7)-(8) (where constant γ is defined). For the last transition we define

$$k = \min\left\{\frac{2}{4\beta^2 d}\left(\frac{1}{2\lambda\beta}\right)^2, \gamma\left(\frac{1}{2\lambda\Lambda\beta^2}\right)^2\right\}.$$

□

C.1.2. A MIXTURE CLASSIFIER

Assume a mixture of two linear classifiers, and let $E(m) = p \cdot E_{\mathcal{D}_1}(m_1) + (1-p) \cdot E_{\mathcal{D}_2}(m_2)$ denote its error score.

Theorem 5. *Keep the assumptions stated in Thm. 4, and assume in addition that $\forall a \in (0,1)$, $\exists \lim\limits_{m\to\infty} \frac{E'(m)}{E(m)}, \lim\limits_{m\to\infty} \frac{E(m)}{E(am)} \lim\limits_{m\to\infty} \frac{E'(m)}{E'(am)}$. Then the error score of a mixture of two linear classifiers is undulating.*

Proof. In each region j of the mixture, Thm. 4 implies that its corresponding error score as defined in Def. 1,

$$E_{\mathcal{D}_j}(m_j) = \mathbb{E}_{\mathbb{X}^{m_j}\sim\mathcal{D}_j^{m_j}}\left[F(m_j)\right]$$

is bounded by an exponentially decreasing function of m_j. Since $E_{\mathcal{D}_j}(m_j)$ measures the expected error over all samples of size m_j, it can also be shown that $E_{\mathcal{D}_j}(m_j)$ is monotonically decreasing with m_j. From the separability assumption, $\lim_{m_j\to\infty} E_{\mathcal{D}_j}(m_j) = 0$. Finally, since $E(m)$ is a linear combination of two such functions, it also has these properties. We conclude from Cor. 1 that the error score of a mixture of two linear classifiers is undulating. □

C.2. 1-NN Classifier

If the training sample size m is small, our analysis shows that under certain circumstances, it is beneficial to prefer sampling from a region R where $E_{\mathcal{D}_R}(m) < E_{\mathcal{D}_{\Omega\setminus R}}(m)$. We now show that the set of densest points has this property.

To this end, we adopt the one-Nearest-Neighbor (1-NN) classification framework. This is a natural framework to address the aforementioned question for two reasons: (i) It involves a general classifier with desirable asymptotic properties. (ii) The computation of both class density and 1-NN is governed by local distances in the input feature space.

To begin with, assume a classification problem with k classes that are separated by at least ρ. More specifically, assume that $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$, if $\|\boldsymbol{x}' - \boldsymbol{x}\| \leq \rho$ then $y' = y$. Let $B_v(\boldsymbol{x}_i, r)$ denote a ball centered at $\boldsymbol{x}_i \in \mathbb{R}^d$, with radius smaller than r and volume v. For $\mathbf{X} = (\boldsymbol{x}, y)$, let $f_{\mathcal{D}}(\mathbf{X})$ denote the density function from which data is drawn when sampling is random (and specifically at test time).

Assume a 1-NN classifier whose training sample is $T =$

$\{\boldsymbol{x}_i, y_i\}_{i=1}^m$, and whose prediction at $\mathbf{X} = (\boldsymbol{x}, y)$ is

$$y = \begin{cases} y_\nu, \;\; \nu = \arg\min\limits_{i\in[m]} \|\boldsymbol{x} - \boldsymbol{x}_i\| & \boldsymbol{x} \in B_v(\boldsymbol{x}_i, \rho) \\ y \sim U(1,k) & \text{otherwise} \end{cases}$$

The error probability of this classifier at $\boldsymbol{x}$ is

$$P(x) = \begin{cases} 0 & \exists \text{ i such that } \boldsymbol{x} \in B_v(\boldsymbol{x}_i, \rho) \\ \frac{k-1}{k} & \text{otherwise} \end{cases}$$

The $0-1$ loss of this classifier is

$$\mathbb{E}_{\mathbf{X}\sim\mathcal{D}}[P(x)] = \frac{k-1}{k} Prob\left[\boldsymbol{x} \notin \bigcup_{i=1}^m B_v(\boldsymbol{x}_i, \rho)\right], \quad (11)$$

where $B_v(\boldsymbol{x}_i, r)$ denotes a ball centered at $\boldsymbol{x}_i \in \mathbb{R}^d$, with radius smaller than r and volume v.

The next theorem states properties of set T which are beneficial to the minimization of this loss:

Theorem 6. *Let A_i denote the event $\{\boldsymbol{x} \in B_v(\boldsymbol{x}_i, r)\}$, and assume that these events are independent. Then we have*

$$L(T) = \frac{k-1}{k}\left[1 - \sum_{i=1}^m f_{\mathcal{D}}(\mathbf{X}_i)v + O(v^2)\right].$$

Proof. Using the independence assumption and (11), and assuming that v is sufficiently small

$$\begin{aligned} L(T) &= \frac{k-1}{k}\left[1 - P\left(\bigcup_{i=1}^m A_i\right)\right] \\ &= \frac{k-1}{k}\left[1 - \sum_{i=1}^m P(A_i)\right] \\ &= \frac{k-1}{k}\left[1 - \sum_{i=1}^m \int_{\mathbf{X}} \mathbb{1}_{\boldsymbol{x}\in B_v(\boldsymbol{x}_i,r)} f_{\mathcal{D}}(\mathbf{X}) d\mathbf{X}\right] \\ &= \frac{k-1}{k}\left[1 - \sum_{i=1}^m f_{\mathcal{D}}(\mathbf{X}_i)v + O(v^2)\right]. \end{aligned}$$

□

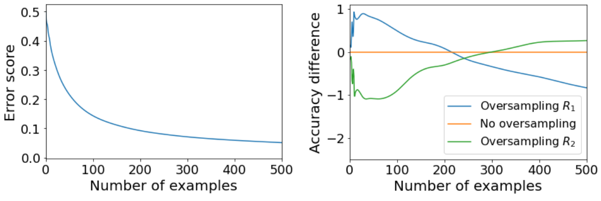
In Thm. 6, we show that if v is sufficiently small, the 0-1 loss is minimized when choosing a set of independent points $\{\mathbf{X}_i\}_{i=1}^m$ that maximizes $\sum_{i=1}^m f_{\mathcal{D}}(\mathbf{X}_i)$. This suggests that the selection of an initial pool of size m will benefit from the following heuristic:

- **Max density:** when selecting a point $\mathbf{X}_i$, maximize its density $f_{\mathcal{D}}(\mathbf{X}_i)$.
- **Diversity:** select varied points, for which the events $\{\boldsymbol{x} \in B_v(\boldsymbol{x}_i, \rho)\}$ are approximately independent.

D. Theoretical Analysis: Visualization

D.1. Mixture of Two Linear Models

We now empirically analyze the error score function of the mixture of two linear classifiers, defined in Section 2.4.1. Each linear classifier is trained on a different area of the support. The data is 100 dimensional, linearly separable in each region. The margin is used to determine the α of the data. The data is chosen such that $p = 0.9, \alpha = 0.2$. The results are shown in Fig. 10.



(a) Error score

(b) Phase transition

Figure 10. (a) The error score $E(m)$ as a function of the number of examples, averaged over $10k$ repetitions. While the error score is not exponential, it could be upper bounded by an exponential function, as analytically shown in Section 2.4.1. (b) The differences in accuracy when over-sampling from either R_1 and R_2 over a random sampling from the data distribution. Although the error score is only proven to be undulating, we can see that in practice it is also SP-undulating.

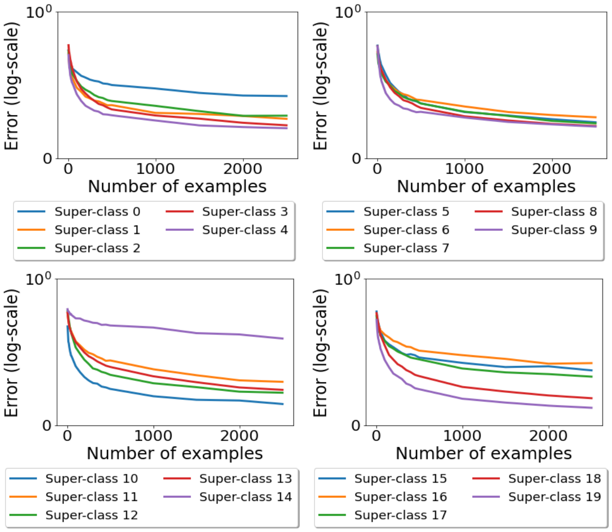
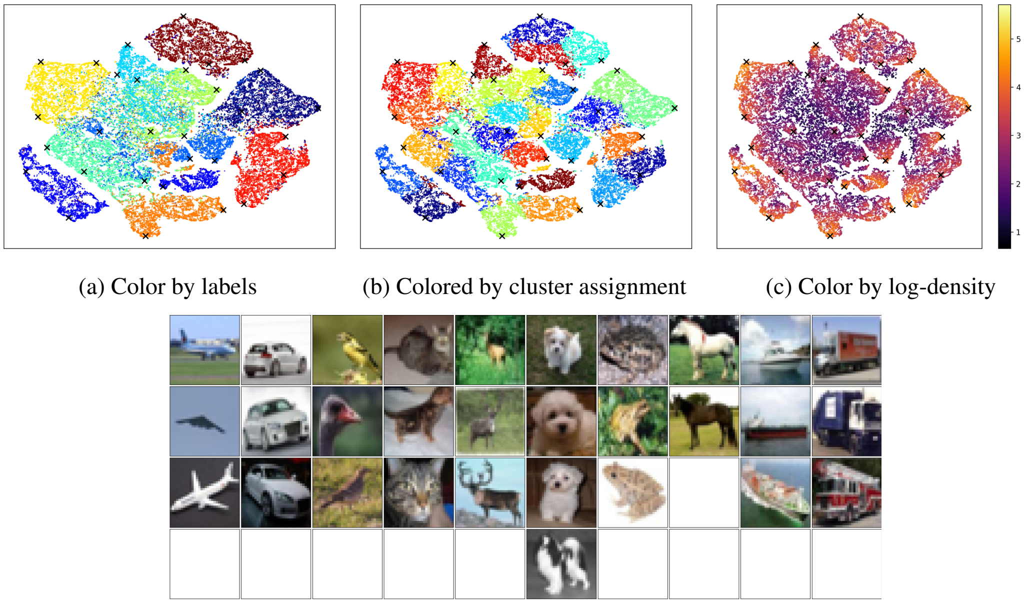


Figure 11. Log-error scores of image classification datasets as a function of the number of training examples. Each score is calculated as $1 - accuracy$, averaged on 100 VGG-16 networks trained on super-classes of CIFAR-100. Each line represents the error of a different super-class. As the log of the error is plotted, it can be seen that in all cases the error scores are monotonic decreasing and can be bounded above by some exponential function, suggesting that often the assumptions in Thm. 3 hold.



(a) Color by labels (b) Colored by cluster assignment (c) Color by log-density

(d) The 30 samples are selected after clustering by SCAN to 30 clusters.

Figure 12. (a)-(c) Visualizing the selection of 30 examples using the SCAN clustering algorithm – examples marked with × are selected for labeling. (d) The selected images, each column represents a different label.

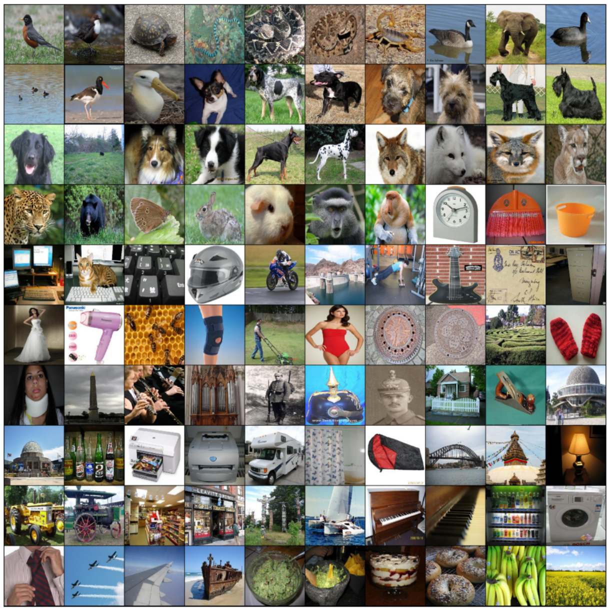


Figure 13. 100 ImageNet-100 examples selected by *TypiClust*.

D.2. Error Scores of Deep Neural Networks

Next, we plot the error scores of deep neural networks on image classification tasks. In all the datasets we evaluated, the error of deep networks as a function of the number of examples drops much faster than an exponential function and therefore can be shown to be undulating. In practice, such error functions are bounded from above by an exponent, and hence are also SP-undulating. To see some examples of error functions of neural networks trained on super-classes of CIFAR-100, refer to Fig. 11.

E. Visualization of Query Selection

Fig. 12 demonstrates the selection of 30 examples from CIFAR-10 using *TypiClust* in greater detail. Recall that TypiClust first clusters the dataset to 30 clusters – using SCAN clustering algorithm. We plot the tSNE dimensionality reduction of the model's feature space, colored in various ways: Fig. 12a shows the tSNE embedding colored by the GT labels. Fig. 12b shows the tSNE embedding colored by the cluster assignment. Fig. 12c shows the tSNE embedding colored by the log density (for better visualization). Examples marked with × are selected for labeling. Fig. 12d shows the selected images. Fig. 13 shows 100 examples selected by *TypiClust* from ImageNet-100.

F. Implementation Details

F.1. Method Implementation Details

Step 1: Representation learning – CIFAR and TinyImageNet. We trained SimCLR using the code provided by Van Gansbeke et al. (2020) for CIFAR-10, CIFAR-100 and TinyImageNet. 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 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.

Step 1: Representation learning – ImageNet. We extracted embedding from the official (ViT-S/16) DINO weights pre-trained on ImageNet. We used the L2 normalized penultimate layer as embedding.

Step 2: Clustering for diversity. We limited the number of clusters when partitioning the data to $max\_clusters$ (a hyperparameter). This parameter was arbitrarily picked as 500 for CIFAR-10 and CIFAR-100 and 1000 for TinyImageNet and ImageNet subsets (other values resulted in similar behavior). This was done for two reasons: (a) prevent over clustering (ending up with clusters that are too small); (b) stabilize the clustering algorithms. The number of clusters chosen is $K = \min(|L_{i-1}| + B, max\_clusters)$.

K-Means. We used scikit-learn KMeans when $K \leq 50$ and MiniBatchKMeans otherwise. This was done to reduce runtime when the number of clusters is large.

SCAN. We used the code provided by SCAN and modified the number of clusters to K. We only trained the first step of SCAN (we did not perform the pseudo labeling step, since it degraded clustering performance).

Step 3: Clustering for diversity. Since we introduced $max\_cluster$, we are no longer guaranteed to have B clusters that don't intersect the labeled set. Moreover, to estimate typicality, we require > 20 samples in every cluster. To solve this, we used $\min\{20, cluster\_size\}$ nearest neighbors. To avoid inaccurate estimation of the typicality, we dropped clusters with less than 5 samples[2].

After all is done, the method adds points iteratively until the budget is exhausted, in the following way: (1) Out of the clusters with the fewest labeled points and of size larger than 5, select the largest cluster. (2) Compute the Typicality of every point in the selected cluster, using $\min\{20, cluster\_size\}$ neighbors. (3) Add to the query the point with the highest typicality.

F.2. Evaluation and Implementation Details

F.2.1. Fully Supervised Evaluation

We used the active learning comparison framework by Munjal et al. (2020). Specifically, we trained a ResNet18 on the labeled set, optimizing using SGD with 0.9 momentum and Nesterov momentum. The initial learning rate is 0.025 and was modified using a cosine scheduler. The augmentations used are random crops and horizontal flips. Our changes to this framework are listed below.

Re-Initialize weights between iterations When training with extremely low budgets, networks tend to produce overconfident predictions. As a result, when querying samples and fine-tuning from the existing model, the loss tends to "spike", which leads to optimization issues. Therefore, we re-initialized the weights between iterations.

TinyImageNet modifications As training did not converge in the original implementation over Tiny-ImageNet, we increased the number of epochs from 100 to 200 and changed the minimal crop side from 0.08 to 0.5. This ensured stable convergence and a more reliable evaluation in AL experiments.

ImageNet modifications ImageNet hyper-parameters were identical to TinyImageNet except for the number of epochs, which was set to 100 due to high computational cost, and the batch size, which was set to 50 to fit into a standard GPU virtual memory.

F.2.2. Linear Evaluation on Self-Supervised Embedding

In these experiments, we also used the framework by Munjal et al. (2020). We extracted an embedding as described in Appendix F.1, and trained a single linear layer of size $d \times C$ where d is the feature dimensions, and C is the number of classes. To optimize this single layer, we increased the initial learning rate by a factor of 100 to 2.5, and as the training time is much shorter, we multiplied the number of epochs by 2.

F.2.3. Semi-Supervised Evaluation

When training FlexMatch, we used the semi-supervised framework by Zhang et al. (2021). All experiments were repeated 3 times. We used the following hyper-parameters when training each experiment:

CIFAR-10. We trained WideResNet-28, for 400k iterations. We used SGD optimizer, with 0.03 learning rate, 64 batch size, 0.9 momentum, 0.0005 weight decay, 2 widen factor, 0.1 leaky slope and without dropout. The augmentations are similar to those used in FlexMatch. The weak augmen-

[2] This limiting case was rarely encountered, as clusters are usually balanced.

tations include random crops and horizontal flips, while the strong augmentations are according to RandAugment (Cubuk et al., 2020).

CIFAR-100. We trained WideResNet-28, for 400k iterations. We used an SGD optimizer, with 0.03 learning rate, 64 batch size, 0.9 momentum, 0.0001 weight decay, 8 widen factor, 0.1 leaky slope, and without dropout. The augmentations are similar to those used in CIFAR-10.

TinyImageNet. We trained ResNet-50, for 1.1m iterations. We used an SGD optimizer, with a 0.03 learning rate, 32 batch size, 0.9 momentum, 0.0003 weight decay, 0.1 leaky slope, and without dropout. The augmentations are similar to those used in FlexMatch.

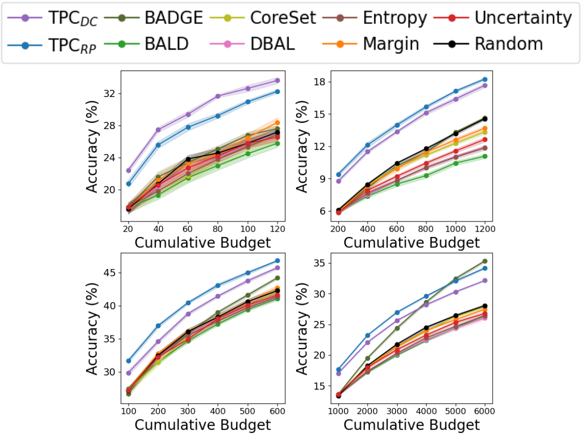
F.3. Ablation studies

Margin by an "oracle" network. In Section 4.3.4, we compute an "oracle" uncertainty measure. When training an oracle, we use a VGG-19 (Simonyan & Zisserman, 2014) trained on CIFAR-10, using the hyper-parameters of the original paper. We calculate the margin of each example according to this network. We note that an AL strategy based on this margin works well in the high-budget regime for both the oracle and the "student" network.

G. Additional Empirical Results

G.1. Supervised Framework

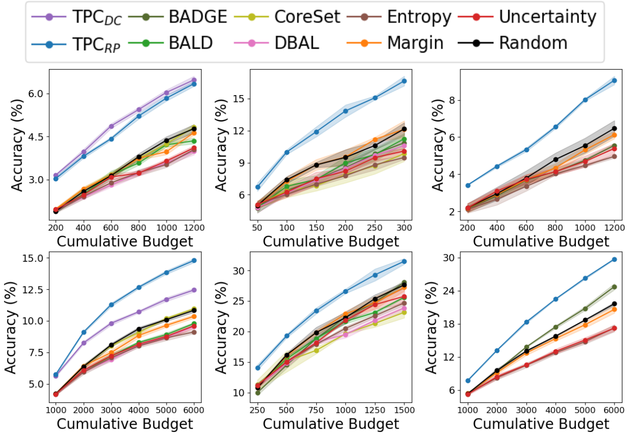
In the main paper, we presented results on 1 and 5 samples per class on average. Fig. 14 shows similar results using additional budget sizes.



(a) CIFAR-10 (b) CIFAR-100

Figure 14. Additional results in the supervised framework, including an average of 2 and 10 examples per class on CIFAR10 and CIFAR100, similarly to Fig. 4.

In Fig. 15 we present results on additional datasets, which include ImageNet-50, ImageNet-100 and TinyImageNet. *TypiClust* outperforms all competing methods on these datasets as well.

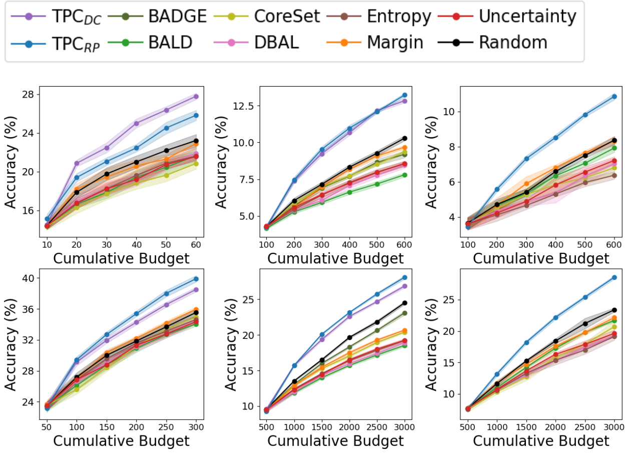


(a) TinyImageNet (b) ImageNet-50 (c) ImageNet-200

Figure 15. Similar to Fig. 4, we report the results on TinyImageNet, ImageNet-50 and ImageNet-200.

G.1.1. Random Initial pool

In Section 4.3.1, we show that even when the initial pool is sampled randomly, *TypiClust* still improves over random selection. Fig. 16 provides additional evidence for this phenomenon, on several datasets and budget sizes.



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

Figure 16. Similar to Fig. 7a, we also report results on CIFAR-100 and ImageNet-100 with an additional budget.

G.1.2. Imbalanced CIFAR-10

Fig. 17 presents results on an imbalanced subset of CIFAR-10, where the number of samples per class decreases exponentially.

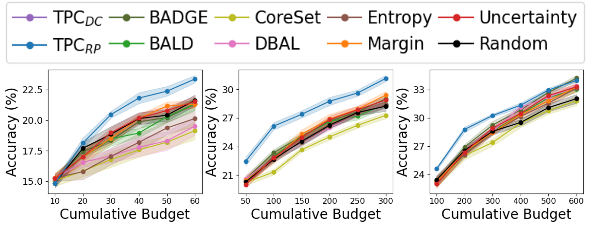
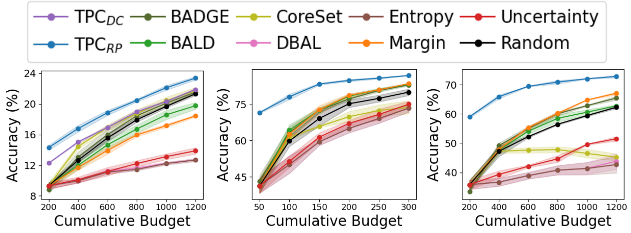


Figure 17. Imbalanced CIFAR-10 results on 3 different budgets.

G.2. Supervised Using Self-Supervised Embeddings

In Fig. 18, we show results using additional datasets of a linear classifier trained over a self-supervised pre-trained embedding. We see that the initial pool selection provides a very large boost in performance – especially with ImageNet-50 and ImageNet-200.



(a) TinyImageNet (b) ImageNet-50 (c) ImageNet-200

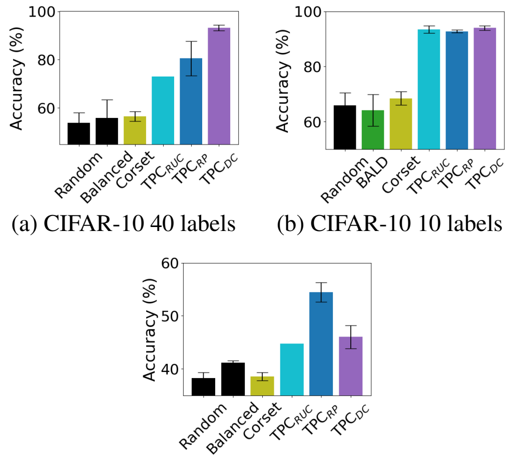
Figure 18. Similar to Fig. 5, we report the linear evaluation results on TinyImageNet, ImageNet-50 and ImageNet-200.

G.3. Semi-Supervised Framework

Finally, below we describe additional experiments to those plotted in Fig. 6, within the semi-supervised framework.

To test the dependency of our deep clustering variant of *TypiClust* on SCAN, we evaluated another variant based on RUC (Park et al., 2021), which is henceforth denoted TPC_{RUC}. We plot its performance on CIFAR-10 and CIFAR-100 in Fig. 19. As RUC is computationally demanding, we fix the number of clusters to the number of classes in the corresponding dataset, and then (when needed) further subcluster the data using K-means to the desired number of clusters. In all the tested settings, TPC_{RUC} surpassed the performance of the random baseline by a large margin, suggesting that using SCAN is not crucial for *TypiClust*, and may be replaced by any competitive clustering algorithm.

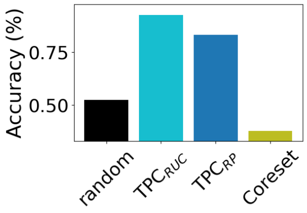
Additionally, we performed experiments with other budgets. In Fig. 19a, we plot the same experiment as Fig. 6a, with a budget of 40 examples on CIFAR-10, seeing similar results.



(a) CIFAR-10 40 labels (b) CIFAR-10 10 labels

(c) CIFAR 100

Figure 19. Similar to Fig. 6, using the TPC_{RUC} variant of *TypiClust*. (a) 40 labels on CIFAR-10, (b) 10 labels on CIFAR-10, (c) 300 labels on CIFAR-100



(a) CIFAR-10 20 labels

Figure 20. Similar to Fig. 6, using Semi-MMDC instead of FlexMatch. We train 20 labels on CIFAR-10, observing a large performance gain when using *TypiClust* to perform the initial selection of the labeled data.

To verify that the observed performance boost is not unique to FlexMatch, we repeat the same experiments with another competitive semi-supervised learning method, Semi-MMDC (Lerner et al., 2020). Using the code provided by Lerner et al. (2020), and following the exact training protocol, we train Semi-MMDC using 20 labels on CIFAR-10. Similarly to the results on FlexMatch, we report a significant increase in performance when training on examples chosen by *TypiClust*, see Fig. 20.

We note that in all the experiments we performed within the low budget regime, *TypiClust* always surpassed the random baseline by a large margin.