Smooth Sailing: Improving Active Learning for Pre-trained Language Models with Representation Smoothness Analysis

Josip Jukić **Jan Šnajder**
University of Zagreb, Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab, Croatia
{josip.jukic, jan.snajder}@fer.hr

Abstract

Developed to alleviate prohibitive labeling costs, active learning (AL) methods aim to reduce label complexity in supervised learning. While recent work has demonstrated the benefit of using AL in combination with large pre-trained language models (PLMs), it has often overlooked the practical challenges that hinder the effectiveness of AL. We address these challenges by leveraging representation smoothness analysis to ensure AL is *feasible*, that is, both effective and practicable. Firstly, we propose an early stopping technique that does not require a validation set – often unavailable in realistic AL conditions – and observe significant improvements over random sampling across multiple datasets and AL methods. Further, we find that task adaptation improves AL, whereas standard short fine-tuning in AL does not provide improvements over random sampling. Our work demonstrates the usefulness of representation smoothness analysis for AL and introduces an AL stopping criterion that reduces label complexity.[1]

1 Introduction

The notorious data hungriness of deep learning models emphasizes the importance of efficient and effective label acquisition. However, the labeling process is often tedious and expensive, ultimately slowing the development of labeled datasets and resulting in subpar models. Evolved out of a practical necessity, *active learning* (**AL**; Cohn et al., 1996; Settles, 2009) is a special family of machine learning algorithms designed to reduce *label complexity* – the number of labels that a learning algorithm requires to achieve a given performance (Dasgupta, 2011) – and thus minimize labeling costs. An AL method aims to select the most informative examples, which can be particularly useful when unlabeled data are abundant, but the labeling is costly or requires substantial expertise.

The striking success of deep learning has motivated the use of traditional AL techniques for training deep neural networks (DNNs) and the development of novel AL methods suited specifically to DNNs. In natural language processing (NLP), AL has been shown to outperform a random selection of examples in many NLP tasks (Zhang et al., 2017; Siddhant and Lipton, 2018; Ikhwantri et al., 2018). Before the widespread adoption of large pre-trained language models (PLMs), a typical AL approach to training deep models was to use task-specific neural models trained from scratch in each AL step (Kasai et al., 2019; Prabhu et al., 2019). Since PLMs fine-tuned to downstream tasks outperform standard neural models, PLMs have supplanted most of them, and researchers have begun to investigate the feasibility of AL for PLMs (Ein-Dor et al., 2020; Schröder et al., 2022). Recent work in AL experimented with several training regimes, such as PLM adaptation and specific fine-tuning techniques (Yuan et al., 2020; Margatina et al., 2022). In particular, task-adaptive pre-training (**TAPT**) has emerged as a cost-effective method for performance improvement complementary to AL (Howard and Ruder, 2018). TAPT uses additional pre-training on the unlabeled training set via masked language modeling and self-supervision. In theory, combining AL with adapted PLMs should produce greater reductions in label complexity than either of the methods in isolation. However, since research on combining AL with PLMs is still in its infancy, whether it can work consistently better than random selection in realistic conditions remains an open question.

One of the challenges in combining AL and PLMs is that, although AL is conceptually simple and promises efficiency gains, there are a host of practical challenges in deploying it in realistic conditions (Attenberg and Provost, 2011; Lowell

[1]Our code is available at https://github.com/josipjukic/al-playground

et al., 2019). The situation is further aggravated by the fact that most AL research overlooks these challenges and resorts to unrealistic evaluation setups and resources. One of the most pervasive problems stems from using a hold-out set during training (e.g., a validation set for regularization by early stopping). In real applications, hold-out sets are unlikely to be available, as building them would require additional labeling effort the AL is meant to reduce in the first place. Another major problem is the flawed evaluation of AL methods: typically, an AL method is compared against random selection as the baseline, but the two training regimes are not kept identical, which confounds the measured effect of AL. In addition to the above-mentioned problems, there is the important practical question of when to stop the acquisition of labels, i.e., how to define the AL *stopping criterion*.

AL methods rely highly on the *acquisition model* (the underlying model used for selecting examples). Therefore, it is important to maintain good generalization properties of the acquisition model, which can be analyzed using representation smoothness. Recently, functional space theory has emerged as a valuable tool for analyzing generalization properties and expressivity of DNNs (Yarotsky, 2017; Suzuki, 2019). In particular, the *Besov space*, a general function space that can capture spatial inhomogeneity, appears convenient for such analyses (Suzuki and Nitanda, 2021).

In this work, we address the practical challenges of AL. First, we systematically evaluate the **feasibility**, where we consider an AL method to be feasible if it is both *practicable* (achievable in realistic conditions) and *effective* (consistently outperforms random selection). Concretely, we explore different learning regimes in AL on various NLP classification tasks without a validation set that is unavailable in most real-world labeling campaigns. Motivated by the effectiveness of TAPT for PLMs (Gururangan et al., 2020), we explore how TAPT combines with AL in the low-resource setup. Secondly, we leverage the representation smoothness of PLM layers in the Besov space to improve AL effectiveness. In particular, we develop *Besov early stopping*, an early stopping regularization technique that does not require a validation set, and we show that it consistently improves the model performance and reduces the variance of results for all AL methods we consider. Moreover, Besov early stopping shows promise as a surrogate for a validation set in zero- and few-shot setups for regular training without AL. We also utilize representation smoothness to develop a stopping criterion based on the smoothness of AL samples to minimize label complexity. Our experiments show a reduction in label complexity for PLMs across five NLP datasets and five AL methods. In addition, building on the idea that representation smoothness is relevant for AL, we complement our experiments with a novel AL method based on the norm of representation gradients. Both the proposed method and the existing AL methods consistently outperform random selection on PLMs with TAPT, which supports the recent findings that the training regime is more important than the choice of the AL method (Margatina et al., 2022).

Our contributions can be summarized as follows: (1) we conduct a systematic evaluation of AL methods for large PLMs and show that AL is feasible, i.e., it consistently outperforms random selection under realistic conditions, (2) we analyze the smoothness of the representation space of PLMs in AL and propose an early stopping technique that improves AL performance and stabilizes the results, (3) we discover patterns in the representation smoothness of AL samples, which we use for an effective AL stopping criterion, and (4) we introduce a representation-based AL method, competitive with other state-of-the-art AL strategies. Our results demonstrate that AL with PLMs is feasible. Even more importantly, the results indicate that representation smoothness analysis can be leveraged to improve model training in general and the effectiveness of AL in particular, opening new avenues for further research.

2 Related Work

Our work builds on several strands of research, including practical challenges in AL, combining AL with PLMs, and different training setups for AL acquisition models.

Practical challenges in AL. Despite the success of AL for many NLP tasks, studies have identified a number of practical challenges hindering the broader deployment of AL (Attenberg and Provost, 2011; Lowell et al., 2019). The most obvious problem is the unavailability of a labeled validation set, an essential resource in model training typically used for hyperparameter optimization and regularization via early stopping. Moreover, in realistic AL conditions, a labeled test set is also unavailable,

making a held-out evaluation of the underlying model's quality impossible. Previous work mostly used model confidence or training error stability to evaluate the acquisition model and derive an AL stopping criterion based on that estimation (Vlachos, 2008; Bloodgood and Vijay-Shanker, 2009; Zhu et al., 2010; Ishibashi and Hino, 2021). However, these criteria have not been widely adopted as they often require tuning for specific datasets and tasks. We mitigate this by developing a task-agnostic AL stopping criterion that detects the points of the largest reduction in label complexity compared to random selection.

AL with PLMs. Only recently have large PLMs been coupled with AL. Early work concentrated mainly on the Transformer architecture (Vaswani et al., 2017) utilizing a simple training setup. More concretely, the predominant approach was to use a standard fine-tuning technique with a fixed number of training epochs, fine-tuning the model from scratch in each AL step (Ein-Dor et al., 2020; Margatina et al., 2021; Shelmanov et al., 2021; Karamcheti et al., 2021; Schröder et al., 2022). However, Mosbach et al. (2021) and Zhang et al. (2021) showed that fine-tuning in low-resource setups (scenarios with little training data) tends to be very unstable, especially when training for only a few epochs. This instability poses a serious issue, as AL often implies a low-resource setting. Moreover, fine-tuning is often sensitive to weight initialization and data ordering (Dodge et al., 2020). This instability of PLM fine-tuning also makes the AL results unstable. We address the instability issue by proposing an early stopping technique without a validation set, and we show that combining PLMs with AL is feasible.

AL training regimes. AL research took a turn from standard fine-tuning of pre-trained models to explore different training regimes and how to use them in combination with AL methods. For example, Grießhaber et al. (2020) explored how to efficiently fine-tune Transformers with AL by freezing the network's layers. Similarly, Yuan et al. (2020) explored self-supervised language modeling to estimate example informativeness for cold-start active learning. Motivated by the general success of TAPT (Gururangan et al., 2020), Margatina et al. (2022) showed that AL outperformed random sampling for PLMs with TAPT, albeit using a validation set. Similarly, Yu et al. (2022) developed a self-training approach for active learning with the addition of weighted clustering. While some training regimes seem promising for AL, the outstanding question is which regimes can consistently outperform random selection. Furthermore, considering what resources are realistically available during training, the primary concern is whether we can apply these training regimes in realistic conditions.

3 Representation in Besov Space

Due to their remarkable flexibility and adaptivity, deep learning models have gained significant traction. To explain these phenomena, researchers have leveraged function space theory to develop approximation and estimation error analysis (Yarotsky, 2017; Suzuki, 2019). Our work relies on a particular type of analysis based on the theory of Besov spaces.

3.1 Besov space

It has been shown that the expressive power of DNNs can be analyzed by specifying the target function's property such as **smoothness** (Petersen and Voigtländer, 2018; Imaizumi and Fukumizu, 2019), i.e., the number of orders of continuous derivatives it has over some domain. *Besov space* has proven to be especially convenient for such analyses, as it allows spatially inhomogeneous smoothness with spikes and jumps, which we often encounter in high-dimensional deep learning. In Besov spaces, the approximation error (expressivity)[2] and estimation error (generalizability)[3] depend on the properties of the representation space (Suzuki and Nitanda, 2021). Given these theoretical connections, representation space analysis can steer toward better generalization properties.

3.2 Besov smoothness index

We briefly describe the mathematical apparatus of the Besov space analysis, adopted with slight modifications from (Suzuki, 2019; Suzuki and Nitanda, 2021). Let $\Omega \in \mathbb{R}^d$ be a domain of functions. For a function $f : \Omega \to \mathbb{R}$ with a defined p-norm in L_p (space of measurable functions with finite p-norm) and seminorm $|f|$ defined by $x \mapsto |f(x)|$, we define $\|f\|_p := \|f\|_{L^p(\Omega)} := (\int_\Omega |f|^p dx)^{\frac{1}{p}}$

[2]The approximation error refers to the distance between the target function and the closest neural network function of a given architecture.

[3]Estimation error refers to the distance between the ideal network function and an estimated network function.

for $0 < p < \infty$. For $p = \infty$, we define $\|f\|_\infty := \|f\|_{L^\infty(\Omega)} := \sup_{x\in\Omega} |f(x)|$.

Definition 1 (Smoothness modulus)**.** *For a function $f \in L^p(\Omega)$, $p \in (0, \infty]$, $t \in (0, \infty)$, $h \in \mathbb{R}^d$, and $r \in \mathbb{N}$, the r-th modulus of smoothness of f is defined by*

$$w_{r,p}(f, t) := \sup_{\|h\|_2 \le t} \|\Delta_h^r(f)\|_p,$$

where $\Delta_h^r(f)$ is the forward difference operator of the r-th order defined as $\Delta_h^r(f)(x) := \sum_{i=0}^r \binom{r}{i}(-1)^{r-i} f(x + ih)$ for $[x, x + rh] \in \Omega$, and 0 otherwise.

Definition 2 (Besov space ($\mathcal{B}^\alpha_{p,q}$))**.** *For $0 < p, q \le \infty$, $\alpha > 0$, $r := \lfloor\alpha\rfloor + 1$, let the seminorm of the Besov space $\mathcal{B}^\alpha_{p,q}$ be*

$$|f|_{\mathcal{B}^\alpha_{p,q}} := \left(\int_0^\infty \left(t^{-\alpha} w_{r,p}(f, t) \right)^q \frac{dt}{t} \right)^{\frac{1}{q}} \quad (1)$$

*for $q < \infty$. Let $|f|_{\mathcal{B}^\alpha_{p,q}} = \sup_{t>0} t^{-\alpha} w_{r,p}(f, t)$ for $q = \infty$. **Besov smoothness index** of f is determined as the maximum index α for which the Besov seminorm is finite.*

Intuitively, the Besov smoothness index (*Besov smoothness* for short) quantifies the properties of DNN's representation space. More specifically, a higher index indicates higher smoothness. Because the calculation of Besov smoothness (more precisely, the integral in (1)) is intractable, we have to rely on approximations. Elisha and Dekel (2016, 2017) proposed wavelet decomposition of a random forest (RF) for approximating Besov smoothness. Wavelet decomposition of the RF establishes an order of importance of the RF nodes, while RF uses the embedded representations of an arbitrary DNN as features. For classification problems, we can normalize the inputs to $[0, 1]$ and transform the class labels into vectors in the $\mathbb{R}^{L-1}$ space by assigning each label to a vertex of a standard simplex, where L is the number of classes. This gives us the k-th layer of a neural network as a function $f_k : [0, 1]^d \to \mathbb{R}^{L-1}$. For a random forest consisting of J estimators, Elisha and Dekel (2017) proceeded by approximating the errors of each estimator $\mathcal{T}_j$ with M most important wavelets. The error function (with $r = 1$, $p = 2$) is estimated as $\sigma_M \sim c_k M^{-\alpha_k}$. Numerically, we can use an approximation $\log(\sigma_m) \sim \log(c_k) - \alpha \log(m)$, $m = 1, \ldots, M$, and find c_k and α_k through least squares, where α_k is the estimate for the Besov smoothness of f_k, i.e., the k-th layer of a DNN.

3.3 Representation smoothness

Analyzing the Besov smoothness of DNNs can unveil their representation geometry. DNNs should benefit from smoother representations, as they help the model avoid overfitting. Intuitively, "well-learned" representations will exhibit high Besov smoothness. When we decompose a PLM into wavelets sorted by relevance and use the Besov smoothness approximation described in Section 3.2, smoother representations achieve lower generalization errors with fewer wavelets.

Another relevant phenomenon for representation smoothness analysis is that the individual layers of DNNs specialize in different features. In particular, earlier layers tend to learn *generalization* features, while the deeper layers are more prone to *memorization* (Stephenson et al., 2021; Baldock et al., 2021). Following these insights, we propose using Besov smoothness to inspect the generalization properties of PLMs through the prism of layer-wise representation geometry. We hypothesize PLMs should benefit more from smoother representations in earlier layers, and we propose methods to enforce learning such representations during training.

4 Preliminaries

In this section, we describe our experimental setup, detailing the datasets, models, AL methods, and evaluation metrics.

4.1 Datasets

We select three different single-text classification tasks commonly used in the AL literature. The datasets vary in size, number of classes, and complexity, allowing for a nuanced study of AL methods. To extend our analysis to similar datasets with different levels of complexity, we also add binary versions of the multi-class tasks. In total, we work with five datasets (cf. Appendix, Table 3): (1) the question type classification dataset (**TREC-6**; Li and Roth, 2002); (2) the corresponding binary version **TREC-2** with only the two most frequent classes (*Entity* and *Human*); (3) the subjectivity dataset **SUBJ** of Pang and Lee (2004), which classifies the movie snippets as subjective or objective and is often used in AL benchmarks; (4) the AG's News classification dataset **AGN-4** of Zhang et al. (2015); and (5) its binary version, **AGN-2**, often used in the AL literature, with two categories (*World* and *Sports*) out of four.

4.2 Models

We focus on large PLMs and include two representatives of the Transformer family, each using a different pre-training paradigm. Specifically, we experiment with BERT (Devlin et al., 2019), which uses a generative pre-training approach via masked language modeling, and ELECTRA (Clark et al., 2020), which relies on discriminative training to detect corrupted tokens induced by a small generator network. For both models, we leverage their widely used *base* variants from the Hugging Face library (Wolf et al., 2020), which consist of 12 layers.

4.3 Active learning methods

We consider six sampling strategies, including random selection, which serves as a baseline. The other five strategies are AL methods from different families.

Random selection (**RND**) selects instances uniformly from the unlabeled pool.

Maximum entropy (**ENT**; Lewis and Gale, 1994) comes from the family of *uncertainty* strategies. The method queries instances where the model is least certain, according to the criterion of maximum entropy of the prediction output.

Monte Carlo dropout (**MC**; Gal and Ghahramani, 2016) is similar to ENTROPY, but relies on the stochasticity of forward passes with dropout layers (Srivastava et al., 2014) to estimate the entropy for a given instance.

Core-set (**CS**; Sener and Savarese, 2018) promotes instance diversity by leveraging the learned representations of the acquisition model. The method aims to minimize the distance between an example in the unlabeled set and its most similar counterpart in the labeled subset.

Discriminative active learning (**DAL**; Gissin and Shalev-Shwartz, 2019) frames active learning as a classification of whether a particular instance is labeled or not to make the labeled and unlabeled sets indistinguishable. Specifically, DAL queries instances that are most likely to be in the unlabeled subset according to a trained classifier.

Representation gradients (**RG**) is a novel AL strategy we propose in this work. Similar to methods from (Huang et al., 2016; Ash et al., 2019), RG selects instances based on gradient information from the representation space. However, unlike other gradient-based methods, RG is much less computationally demanding and, therefore, suitable for resource-limited studies and realistic conditions. The method computes the mean representation gradient with respect to the embedded inputs and selects the instances with the largest gradient norm. Formally, with $\bar{\mathbf{h}}$ as the mean representation, the RG's selection criterion is $\operatorname{argmax}_{\mathbf{x}\in\mathcal{U}} \left\|\partial_{\mathbf{x}}\bar{\mathbf{h}}\right\|_2$,

where $\mathcal{U}$ denotes the unlabeled set. The intuition behind RG is that the locally sharp instances in the representation space of the underlying model, i.e., the ones with large gradient norms, surprise the model the most and thus will contribute the most to a reduction in label complexity.

In our experiments, we select 50 new examples in each step of each AL experiment, using 100 examples for the warm start (randomly sampled labeled data to kick-start the model). We set the labeling budget to $1,000$ instances for easier datasets (TREC-2, AGN-2, and SUBJ) and $2,000$ instances for harder datasets (TREC-6 and AGN-4).

4.4 Evaluation

To evaluate the entire AL process, we use the area under the performance curve (**AUC**). Each step corresponds to the classification performance in terms of the F_1 score of a model trained with a certain number of labeled examples. We advocate using AUC complementary to the AL curves, as we believe it is a good approximation of AL feasibility as a summary numeric score. Since we use different training regimes in our experiments, we compare each AL strategy to random selection within the same training regime to isolate the effects of AL. Additionally, we introduce a metric to measure the direct practical gains of AL by estimating the reduction in label complexity of AL compared to random selection. For a given AL step, we compute the number of additional labels required to achieve the same performance with random selection, thus estimating the number of labels one saves when using AL. We refer to this metric as *label complexity reduction* (**LCR**).

5 Improving Active Learning

In this section, we first look into the representation geometry of PLMs by means of representation smoothness analysis. Then, we link our findings to devise a smoothness-based early stopping technique that does not require a validation set. We explore the effects of our method in different training regimes and provide a systematic evaluation of AL for PLMs in the low-resource setup.

5.1 Representation smoothness analysis

We empirically test the characteristics of Besov smoothness of PLMs. In particular, we compare the representation smoothness of PLMs in three different training regimes: (1) **short training** (ST), where models were trained for 5 epochs, (2) **extended training** (ET), where models were trained for 15 epochs, and (3) model adaptation with TAPT (cf. Appendix A.5 for details) followed by an extended training for 15 epochs (ETA). We computed the smoothness of PLM layers during training, averaged across AL steps. In each AL step, we fine-tuned the model anew.

Performance-wise, ETA yields better results than ET and ST (Table 1). Moreover, AL in the ST regime does not yield improvement over random sampling. Figure 1 shows the layer-wise smoothness for the three mentioned regimes with the addition of the overfitting regime, where we purposefully overfitted the acquisition model in each AL step by training the model for 100 epochs. In the ST regime, we observe a monotonic increase in smoothness as we progress through layers, while the smoothness in ET peaks before the last few layers. The shift of the smoothness peak is even more pronounced for TAPT with extended training. In overfitted models, we observe a flat distribution of smoothness across layers. We observe that better performance and effective AL come with a shift in smoothness distribution towards earlier layers, as displayed in ET and ETA regimes. We hypothesize that, in the low-resource setup, the deeper layers exhibit higher smoothness in the ST regime because they are prone to *heuristic memorization* – DNN relies on spurious artifacts (shortcuts) that are correlated with a target label (Bansal et al., 2022) – which may cause the model to perform poorly.

5.2 Besov early stopping

In the AL loop, the effect of selecting an acquisition model with poor generalization properties

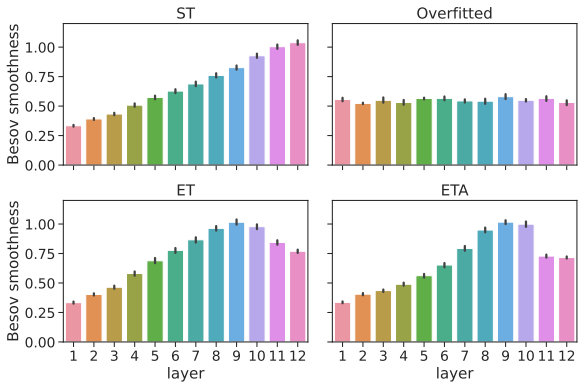


Figure 1: Besov smoothness of PLM layers for different training regimes. The scores are normalized (between 0 and 1 per layer) and averaged across datasets, models, and AL methods. The black error bars represent the standard deviation. We note that the deviation is small, indicating similar behavior across different datasets, models, and AL methods.

propagates through the AL steps. To ensure the effectiveness of AL, regularization by early stopping is often used to pre-empt overfitting in order to retain good generalization properties. However, since a validation set is often unavailable in realistic conditions, using it for early stopping renders AL impracticable. However, feasible AL needs to be both effective and practicable.

The above empirical findings on smoothness distribution across the PLM layers for the different training regimes motivate an early stopping heuristic based on representation smoothness without a validation set. We propose **BEAST** (**B**esov **ea**rly **st**opping), where we proceed with the training as long as the Besov smoothness distribution skews toward earlier layers. We define the stopping point as the epoch where the distribution skewness[4] fails to increase, i.e., when the peak of the representation smoothness ([CLS] token) fails to shift towards earlier layers for two consecutive epochs. We revert the model to the last epoch where this effect is preserved. In this way, we stop the training before the smoothness distribution flattens out, which we observe in overfitted models. We experiment with two more training regimes: ET$^{\mathcal{B}}$ and ETA$^{\mathcal{B}}$, which are just ET and ETA with BEAST.

We compare BEAST to the approaches without early stopping, where we chose the models from the last epoch. Our experiments show the difference in AL performance across different training

[4]We compute the layer-wise smoothness skewness as the Fisher-Pearson coefficient of skewness.

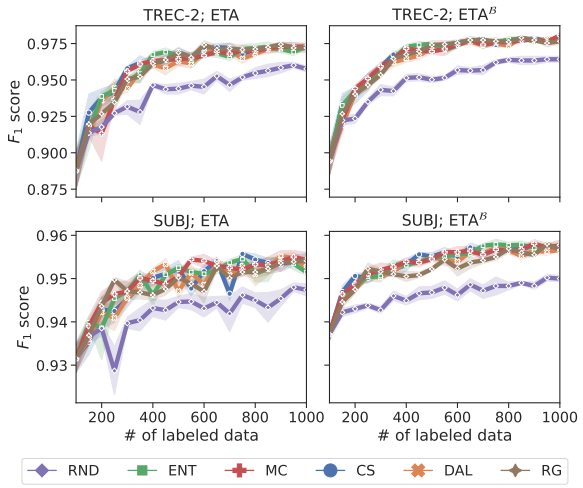


Figure 2: Active learning performance curves for BERT in terms of F_1 score. Random sampling (purple rhombs) serves as a baseline. For the sake of space, we show the results on a subset of datasets for BERT and regimes ETA and ETA$^{\mathcal{B}}$ as we obtained similar results for other configurations (cf. Figure 4 in Appendix). The results are averaged over five runs. The confidence intervals represent the standard deviation. Best viewed on a computer screen.

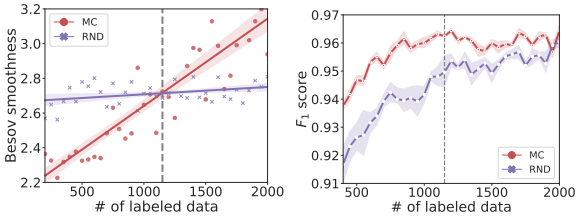
regimes. Figure 2 shows the trend of AL curves through the steps, and Table 1 provides more comprehensive comparisons with AUC as the aggregated measure of AL effectiveness. We can observe that AL coupled with ST performs poorly, and AL fails to outperform random sampling (sometimes even worse than random sampling). The ET regime generally improves performance, with AL sometimes outperforming random selection. ETA and ETA$^{\mathcal{B}}$ further improve performance over random sampling for every AL method on every dataset we used. For BERT, the difference between AL and random sampling is statistically significant in 22 out of 25 cases with ETA and in all 25 cases with ETA$^{\mathcal{B}}$. More importantly, ET$^{\mathcal{B}}$ and ETA$^{\mathcal{B}}$ outperform their counterparts without BEAST and reduce the variance of the results (cf. Appendix, Table 6). We support the hypothesis that the choice of the AL method is not as important as the training regime, as we achieve similar results for every method when AL outperforms random selection. TAPT works across the board, improving AL performance on all five datasets. With the addition BEAST, we achieve feasible AL, making it both practicable and effective. On top of that, even with random sampling, BEAST consistently yields higher scores than the model from the last epoch, showing benefits even for regular fine-tuning without AL.

		RND	ENT	MC	CS	DAL	RG
TREC-2	ST	.875	.873	.883	.881	.889	.879
	ET	.912	.932†	.934†	.929	.931	.931
	ET$^{\mathcal{B}}$	.925	.942†	.942†	.939†	.940†	.938
	ETA	.941	.959†	.957†	.960†	.957†	.958†
	ETA$^{\mathcal{B}}$	.949	**.966**†	.965†	.965†	.964†	.965†
SUBJ	ST	.896	.892	.885	.901	.898	.892
	ET	.920	.922	.922	.925	.925	.920
	ET$^{\mathcal{B}}$	.928	.931	.932	.932	.933	.930
	ETA	.942	.949†	.950†	.949†	.949†	.948
	ETA$^{\mathcal{B}}$	.946	**.954**†	**.954**†	**954**†	.953†	.952†
AGN-2	ST	.923	.942†	.941†	.922	.941†	.942†
	ET	.960	.969	.970†	.965	.967	.969
	ET$^{\mathcal{B}}$	.967	.974†	.975†	.972	.974†	.975†
	ETA	.974	.981†	.980	.981†	.980	.980†
	ETA$^{\mathcal{B}}$	.977	**.983**†	**.983**†	**.983**†	.982†	.982†
TREC-6	ST	.706	.743†	.749†	.666	.689	.693
	ET	.867	.878	.881†	.867	.878	.867
	ET$^{\mathcal{B}}$	.873	.885	.890†	.873	.882	.875
	ETA	.909	.933†	.931†	.931†	.934†	.930†
	ETA$^{\mathcal{B}}$	.925	.939†	.937†	.936†	**.940**†	.935†
AGN-4	ST	.837	.828	.824	.801	.834	.829
	ET	.869	.869	.871	.871	.880†	.875
	ET$^{\mathcal{B}}$	.875	.877	.878	.879	.886†	.881
	ETA	.891	.905†	.905†	.902†	.906†	.899†
	ETA$^{\mathcal{B}}$	.894†	.908†	.908†	.905†	**.909**†	.903†

Table 1: AUC scores for random sampling and different AL methods across datasets and training regimes for BERT (cf. Appendix, Table 5). The results are averaged over 5 runs with different seeds. **Bold** numbers indicate the best AUC for each dataset. The "†" indicates when the mean AUC of an AL method is significantly different from random sampling (two-sided Man-Whitney U test with $p < .05$, adjusted for family-wise error rate with the Holm-Bonferroni method).

6 Active Sample Smoothness

In Section 5, we analyzed the Besov smoothness of layer representations of PLMs. In this section, we take a step further and examine the smoothness at the instance level. Instead of using the representations on the training set, we computed the Besov smoothness as the average across layers on the unseen (selected but not yet trained on) *active sample*, acquired by the AL method. In contrast to the seen training examples, we argue that the Besov smoothness on unseen examples can be interpreted as the amount of information the model could obtain from that sample. More precisely, the lower the smoothness of an active sample, the more informative it is for the model. In contrast, smooth samples are already well-represented and thus not as resource-effective as their less smooth counterparts.



(a) Sample smoothness (b) AL curve

Figure 3: The relationship between an active sample and random sample smoothness. Figure 3a shows how the smoothness of samples retrieved by AL (red) relates to the smoothness of random samples (violet) with fitted regression lines. The smoothness values are calculated as the average across layers. Figure 3b shows the corresponding AL performance curve. The gray dotted line indicates the intersection of active and random sample smoothness, which signals the beginning of diminishing returns of AL. We show the results for BERT in the ETA$^{\mathcal{B}}$ regime on the TREC-6 dataset as an illustration. We observe very similar patterns in other datasets. Since all of the AL methods display similar behavior in ETA$^{\mathcal{B}}$ regime, we show only MC to avoid clutter (cf. Figure 5 in Appendix for other datasets). Best viewed on a computer screen.

We compare the Besov smoothness of actively acquired samples against random samples. We consistently observe two patterns, showcased by Figure 3. First, the smoothness of random samples is uniform throughout the AL steps. The second pattern occurs in the trend of AL sample smoothness. In the early AL steps, AL sample smoothness is low, indicating sharp representations that require smoothing (by learning). As the AL procedure progresses, the acquisition model improves, and the active samples' smoothness increases. We interpret this as the model slowly consuming the information from the data pool, eventually reaching a state of "information depletion", i.e., a state in which the remaining unlabeled data provides no additional value to the model.

Stopping criterion. Our preliminary analysis of the relationship between the active and random sample smoothness motivates a simple stopping criterion, which we refer to as ALSBI (**a**ctive **l**earning **s**topping by **B**esov **i**ndex). ALSBI aims to detect when AL methods reach information depletion. We terminate the AL process when sample smoothness surpasses the average smoothness of a random sample in two consecutive steps. We disregard the first AL step, as it often takes several steps for the acquisition models to stabilize. Since we cannot compute the smoothness of a random sample (as we query only AL samples) in realistic conditions, we estimate the random sample smoothness on the warm start examples via bootstrapping. This approximation proved stable for 100 examples as the smoothness of a random sample remains stable throughout AL steps. We take the average smoothness of $1,000$ bootstrapped samples of size 50. Table 2 shows that ALSBI yields larger LCR than what one would get on average across AL steps, which supports our preliminary analysis. RG achieves the highest LCR among the tested AL methods, which we believe is due to its compatibleness to ALSBI as both the AL method and the stopping criterion are based on representation smoothness.

		ENT	MC	CS	DAL	RG
ETA	avg	.316	.312	.351	.275	.319
	ALSBI	.447	.401	.511	.282	**.521**
ETA$^{\mathcal{B}}$	avg	.385	.385	.394	.354	.382
	ALSBI	.586	.532	.488	.447	**.645**

Table 2: Average LCR across datasets and models. The scores indicate the proportion of the dataset that needs to be labeled for random sampling to match the performance of the corresponding AL method. ALSBI is compared to an average LCR throughout the AL steps (avg). The results are averaged over 5 runs. Numbers in **bold** indicate the largest LCR for a certain training regime.

7 Conclusion

In our paper, we leverage representation smoothness analysis to improve the effectiveness of active learning (AL). In realistic conditions, we show that AL with pre-trained language models (PLMs) is effective when combined with task adaptation, while standard short fine-tuning often fails. We address the problem of unavailable resources (labeled hold-out sets) by developing the **B**esov **ea**rly **s**topping technique (BEAST) that does not require a validation set. For AL to be feasible, it must be both effective and practicable. BEAST meets both feasibility requirements: it improves AL performance over random sampling and reduces the variance of the performance scores across AL steps (effectiveness) while not requiring additional labeled data (practicability). Moreover, BEAST improves the performance of PLMs even in standard fine-tuning without AL, which makes it potentially useful in zero-shot and few-shot setups where a validation set could also be unavailable. We further show the

usefulness of representation smoothness analysis for AL by devising a simple and effective AL stopping criterion. We corroborate the hypothesis from previous research in that the effectiveness of AL is influenced more by the training regime rather than the AL method itself. We believe that the relationship between PLMs' generalization properties, label complexity, and representation smoothness is an exciting avenue for AL, and we hope our results will motivate further research in that direction.

Limitations

To fully comprehend the significance of our findings, it is necessary to consider the limitations of this study. Firstly, we evaluate only two Transformer-based models on a small number of text classification tasks. Although we used the models with different pre-training paradigms, it is possible that the findings do not generalize across models within the same family. In addition, we used the base variants of BERT and ELECTRA, which both feature 12 layers. Since our early stopping criterion is influenced by the number of layers whose smoothness we approximate, there is a possibility that smoothness would distribute differently for models with more or fewer layers. Another limitation is that we did not investigate these models' performance on tasks other than text classification, and the results may not be generalizable to different types of NLP tasks.

Since there are many different ways to measure the quality of an AL stopping criterion and we only wanted to illustrate the usefulness of smoothness patterns, we only compared the proposed ALSBI method against an average baseline. However, a more comprehensive comparison with other approaches from the literature would provide a better understanding of the merit of our method.

Lastly, we only scratched the surface of different training regimes for PLMs in the context of AL. Many new training regimes are emerging in the field, especially the ones focused on efficiency and modularity. We leave the exploration of these methods for future work.

References

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *CoRR*, abs/1906.03671.

Josh Attenberg and Foster Provost. 2011. Inactive learning? Difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12(2):36–41.

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. 2021. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems*, volume 34, pages 10876–10889. Curran Associates, Inc.

Rachit Bansal, Danish Pruthi, and Yonatan Belinkov. 2022. Measures of information reflect memorization patterns. In *Advances in Neural Information Processing Systems*.

Michael Bloodgood and K. Vijay-Shanker. 2009. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 39–47, Boulder, Colorado. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.

Sanjoy Dasgupta. 2011. Two faces of active learning. *Theoretical Computer Science*, 412(19):1767–1781. Algorithmic Learning Theory (ALT 2009).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Oren Elisha and Shai Dekel. 2016. Wavelet decompositions of random forests - smoothness analysis, sparse approximation and applications. *Journal of Machine Learning Research*, 17(198):1–38.

Oren Elisha and Shai Dekel. 2017. Function space analysis of deep learning representation layers. *arXiv preprint arXiv:1710.03263*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

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

Daniel Grießhaber, Johannes Maucher, and Ngoc Thang Vu. 2020. Fine-tuning BERT for low-resource natural language understanding via active learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1158–1171, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Jiaji Huang, Rewon Child, Vinay Rao, Hairong Liu, Sanjeev Satheesh, and Adam Coates. 2016. Active learning for speech recognition: the power of gradients. *arXiv preprint arXiv:1612.03226*.

Fariz Ikhwantri, Samuel Louvan, Kemal Kurniawan, Bagas Abisena, Valdi Rachman, Alfan Farizki Wicaksono, and Rahmad Mahendra. 2018. Multi-task active learning for neural semantic role labeling on low resource conversational corpus. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 43–50, Melbourne. Association for Computational Linguistics.

Masaaki Imaizumi and Kenji Fukumizu. 2019. Deep neural networks learn non-smooth functions effectively. In *AISTATS*.

Hideaki Ishibashi and Hideitsu Hino. 2021. Stopping criterion for active learning based on error stability. *arXiv preprint arXiv:2104.01836*.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. 2021. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7265–7281, Online. Association for Computational Linguistics.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5851–5861, Florence, Italy. Association for Computational Linguistics.

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

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 21–30, Hong Kong, China. Association for Computational Linguistics.

Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2022. On the importance of effectively adapting pretrained language models for active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 825–836, Dublin, Ireland. Association for Computational Linguistics.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Philipp Christian Petersen and Felix Voigtländer. 2018. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural networks : the official journal of the International Neural Network Society*, 108:296–330.

Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling bias in deep active classification: An empirical study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4058–4068, Hong Kong, China. Association for Computational Linguistics.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

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

Burr Settles. 2009. Active learning literature survey. Computer sciences technical report.

Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, Ekaterina Artemova, Dmitry V. Dylov, and Alexander Panchenko. 2021. Active learning for sequence tagging with deep pre-trained models and Bayesian uncertainty estimates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1698–1712, Online. Association for Computational Linguistics.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. 2021. On the geometry of generalization and memorization in deep neural networks. In *International Conference on Learning Representations*.

Taiji Suzuki. 2019. Adaptivity of deep reLU network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*.

Taiji Suzuki and Atsushi Nitanda. 2021. Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic Besov space. In *Advances in Neural Information Processing Systems*, volume 34, pages 3609–3621. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Andreas Vlachos. 2008. A stopping criterion for active learning. *Computer Speech & Language*, 22(3):295–312.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Dmitry Yarotsky. 2017. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114.

Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. 2022. AcTune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1422–1436, Seattle, United States. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample BERT fine-tuning. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jingbo Zhu, Huizhen Wang, Eduard Hovy, and Matthew Ma. 2010. Confidence-based stopping criteria for active learning for data annotation. *ACM Transactions on Speech and Language Processing (TSLP)*, 6(3):1–24.

	TRAIN	VAL	TEST	TOTAL
TREC-2	1,987	159	486	2,632
SUBJ	7,000	1,000	2,000	10,000
AGN-2	20,000	2,600	5,000	27,600
TREC-6	4,881	452	500	5,833
AGN-4	20,000	7,600	7,600	35,200

Table 3: Dataset sizes by splits. Although we do not use a validation set (VAL) in our experiments, we report its size for completeness. We uniformly subsampled the AGN-2 and AGN-4 datasets for shorter computation time.

	BERT	ELECTRA
TREC-2	32.4	31.1
SUBJ	40.8	39.2
AGN-2	71.4	70.3
TREC-6	68.4	67.1
AGN-4	82.3	75.7

Table 4: Experiment duration in minutes for both models across datasets. We report the average runtime over five different runs and six different sampling methods (five AL methods and random sampling).

A Reproducibility

A.1 Dataset statistics

We report the sizes of the datasets per split in Table 3. The datasets contain mainly texts in English.

A.2 Models

We used base and uncased variants of the Transformer models. Specifically, we used "bert-base-uncased" for BERT and "google/electra-base-discriminator" for ELECTRA. Both models have $109,514,298$ trainable parameters each.

A.3 AL methods

MC We use ten inference cycles to approximate the entropy of the output via Monte-Carlo dropout sampling.

CS We use the [CLS] token representation from the Transformer's penultimate layer. We opt for the greedy method described in the original paper (Sener and Savarese, 2018).

A.4 Preprocessing

We use the same preprocessing pipeline on all datasets for both BERT and ELECTRA. We lowercase the tokens, remove non-alphanumeric tokens and truncate the sequence to 200 tokens.

A.5 Hyperparameters

We used a fixed learning rate of 2×10^{-5} for both models. Additionally, we set the gradient clipping to 1 during training. In the ST regime, we trained the model for 5 epochs and 15 in ET, and ETA. For TAPT, we used masked language modeling with 15% of randomly masked tokens and trained the model via self-supervision for 50 epochs with the learning rate set to 10^{-5}.

A.6 Computing infrastructure

We conducted our experiments on $4\times$ *AMD Ryzen Threadripper 3970X 32-Core Processors* and $4\times$ *NVIDIA GeForce RTX 3090* GPUs with 24GB of RAM. We used *PyTorch* version 1.9.0 and CUDA 11.4.

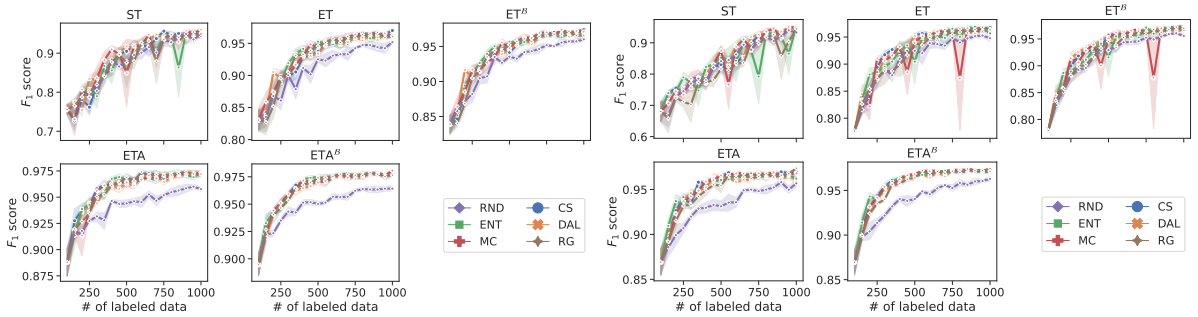
A.7 Average runtime

We report the average runtime of experiments in Table 4. We ran six sampling methods on five datasets for two models and for five different training regimes. Additionally, we repeated each experiment five times with different seeds ($[1, 2, 3, 4, 5]$). In each experiment, we re-train the model 20 times on TREC-2, SUBJ, and AGN-2 up to $1,000$ instances (20 batches of 50 instances), and 40 times on TREC-6 and AGN-4 up to $2,000$ instances (40 batches of size 50). In total, we ran 300 AL experiments.

B Experiments

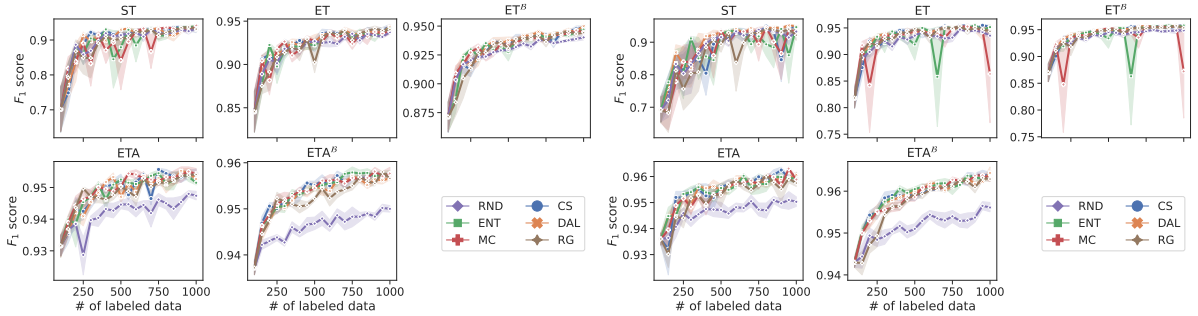
We report the experiments that were omitted from the main part of the paper due to space constraints. Figure 4 shows the active learning performance curves across the used datasets and for both models (BERT and ELECTRA). For the ETA and ETA$^{\mathcal{B}}$ training regimes, we observe a consistent improvement in performance compared to random sampling. We report the results for ELECTRA in Table 5, where we observed similar patterns as with BERT (cf. Table 1 in the main part of the paper). On top of that, our early stopping method reduces the variance of the results compared to other training regimes, as shown in Table 6.

Figure 5 shows the relationship between the Besov smoothness of random and active samples. We report the smoothness of samples for each dataset. We observe a similar pattern, with a rising smoothness of actively acquired samples.



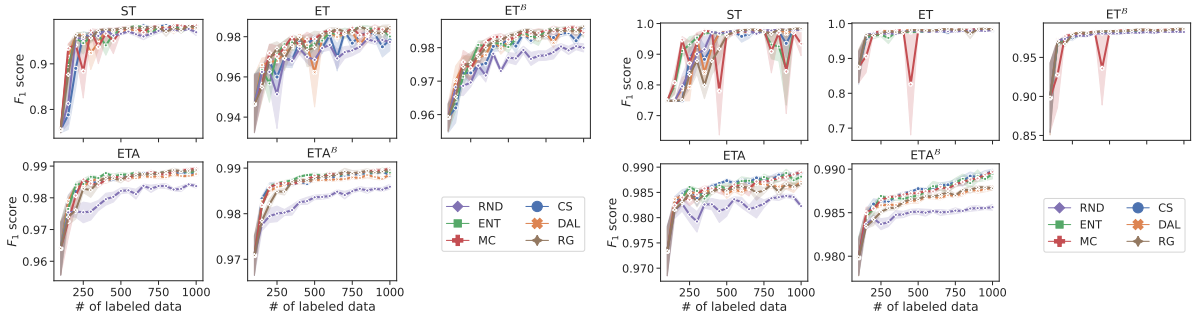
(a) TREC-2; BERT

(b) TREC-2; ELECTRA



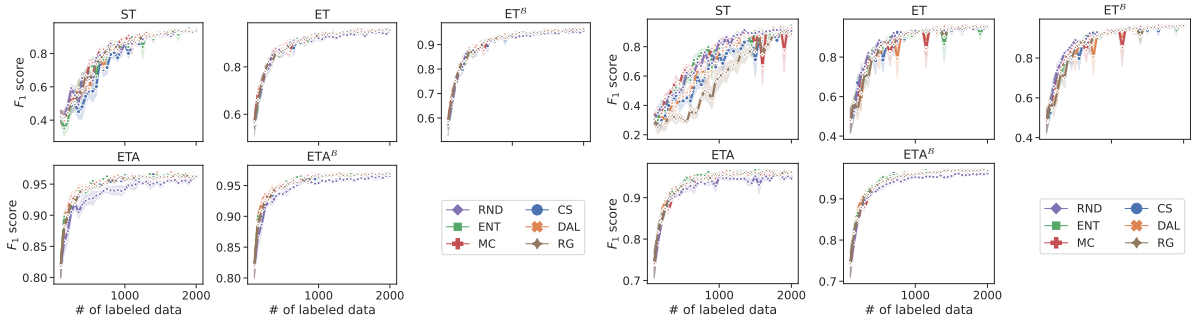
(c) SUBJ; BERT

(d) SUBJ; ELECTRA



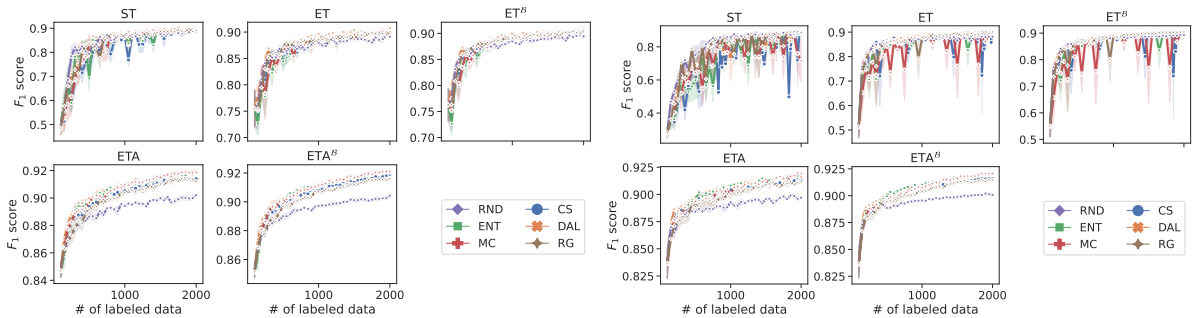
(e) AGN-2; BERT

(f) AGN-2; ELECTRA



(g) TREC-6; BERT

(h) TREC-6; ELECTRA



(i) AGN-4; BERT

(j) AGN-4; ELECTRA

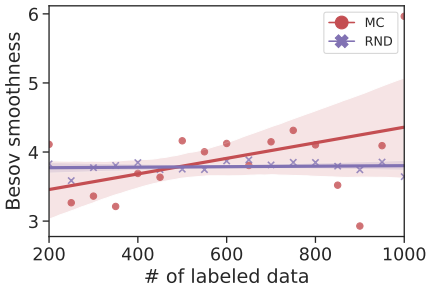
Figure 4: AL performance curves for different training regimes across datasets and models. Random sampling (purple rhombs) serves as a baseline. Best viewed on a computer screen.

		RND	ENT	MC	CS	DAL	RG
TREC-2	ST	.831	.829	.836	.835	.840	.805
	ET	.910	.924	.918	.928	.927	.919
	ET$^{\mathcal{B}}$	.919	.930	.926	.934	.936	.927
	ETA	.932	.953	.953	.953	.951	.949
	ETA$^{\mathcal{B}}$	.939	.959	.958	.959	.956	.956
SUBJ	ST	.880	.872	.870	.871	.898	.860
	ET	.926	.927	.925	.935	.936	.935
	ET$^{\mathcal{B}}$	.938	.937	.934	.944	.946	.942
	ETA	.946	.955	.954	.955	.955	.952
	ETA$^{\mathcal{B}}$	.952	.959	.959	.959	.959	.957
AGN-2	ST	.867	.901	.891	.850	.850	.823
	ET	.963	.963	.954	.963	.966	.965
	ET$^{\mathcal{B}}$	.969	.971	.962	.971	.971	.972
	ETA	.977	.981	.981	.982	.981	.980
	ETA$^{\mathcal{B}}$	.980	.983	.983	.983	.982	.982
TREC-6	ST	.604	.645	.636	.549	.561	.461
	ET	.837	.848	.839	.817	.811	.814
	ET$^{\mathcal{B}}$	.843	.858	.847	.821	.813	.816
	ETA	.897	.917	.905	.905	.905	.901
	ETA$^{\mathcal{B}}$	.906	.925	.917	.915	.914	.911
AGN-4	ST	.793	.706	.713	.688	.755	.750
	ET	.857	.844	.824	.845	.866	.857
	ET$^{\mathcal{B}}$	.868	.855	.836	.857	.874	.866
	ETA	.888	.903	.901	.901	.904	.897
	ETA$^{\mathcal{B}}$	.893	.907	.905	.905	.907	.901

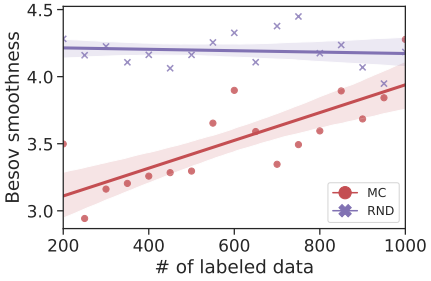
Table 5: AUC for random sampling and different AL methods across datasets and training regimes for ELECTRA. The results are averaged over five runs with different seeds.

	ST	ET	ETA	ET$^{\mathcal{B}}$	ETA$^{\mathcal{B}}$
TREC-2	.0093	.0053	.0045	.0026	**.0022**
SUBJ	.0117	.0045	.0032	.0013	**.0008**
AGN-2	.0100	.0036	.0020	.0009	**.0005**
TREC-6	.0134	.0081	.0074	.0032	**.0027**
AGN-4	.0118	.0048	.0045	.0022	**.0014**

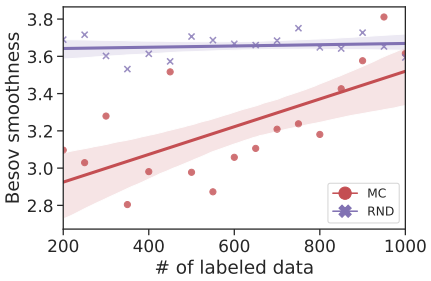
Table 6: Average standard deviation for different training regimes. The results are averaged across models and AL methods. **Bold** numbers indicate regimes with the lowest standard deviation for a particular dataset.



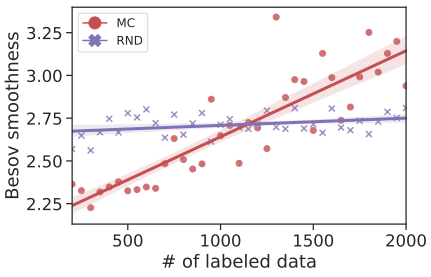
(a) TREC-2



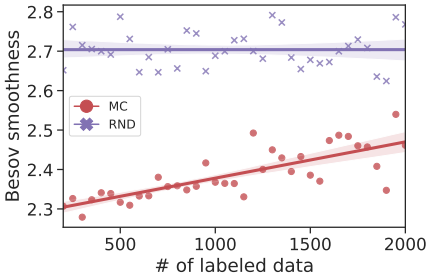
(b) AGN-2



(c) SUBJ



(d) TREC-6



(e) AGN-4

Figure 5: Besov smoothness of actively acquired samples with MC (red) compared to the smoothness of random samples (purple).