Like a Good Nearest Neighbor: Practical Content Moderation and Text Classification

Luke Bates and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP Lab)
Department of Computer Science and Hessian Center for AI (hessian.AI)
Technical University of Darmstadt
www.ukp.tu-darmstadt.de
firstname.lastname@tu-darmstadt.de

Abstract

Few-shot text classification systems have impressive capabilities but are infeasible to deploy and use reliably due to their dependence on prompting and billion-parameter language models. SetFit (Tunstall et al., 2022) is a recent, practical approach that fine-tunes a Sentence Transformer under a contrastive learning paradigm and achieves similar results to more unwieldy systems. Inexpensive text classification is important for addressing the problem of domain drift in all classification tasks, and especially in detecting harmful content, which plagues social media platforms. Here, we propose Like a Good Nearest Neighbor (LaGoNN), a modification to SetFit that introduces no learnable parameters but alters input text with information from its nearest neighbor, for example, the label and text, in the training data, making novel data appear similar to an instance on which the model was optimized. LaGoNN is effective at flagging undesirable content and text classification, and improves SetFit's performance. To demonstrate LaGoNN's value, we conduct a thorough study of text classification systems in the context of content moderation under four label distributions, and in general and multilingual classification settings.[1]

1 Introduction

Text classification is the most important tool for NLP practitioners, and there has been substantial progress in advancing the state-of-the-art, especially with the advent of large, pretrained language models (PLM) (Devlin et al., 2019). Modern research focuses on in-context learning (Brown et al., 2020), pattern exploiting training (Schick and Schütze, 2021a,b, 2022), adapter-based fine-tuning with learned label embeddings (Karimi Mahabadi et al., 2022), and parameter efficient fine-tuning (Liu et al., 2022a). These methods have achieved impressive results on the SuperGLUE (Wang et al., 2019) and RAFT (Alex et al., 2021) few-shot benchmarks, but most are difficult to use because of their reliance on billion-parameter PLMs, pay-to-use APIs, and/or prompting. Constructing prompts is not trivial and may require domain expertise.

One exception to these cumbersome systems is SetFit. SetFit does not rely on prompting or billion-parameter PLMs, and instead fine-tunes a pretrained Sentence Transformer (ST) (Reimers and Gurevych, 2019) under a contrastive learning paradigm. SetFit has comparable performance to more unwieldy systems while being one to two orders of magnitude faster to train and run inference.

An important application of text classification is aiding or automating content moderation, which is the task of determining the appropriateness of user-generated content on the Internet (Roberts, 2017). From fake news to toxic comments to hate speech, it is difficult to browse social media without being exposed to potentially dangerous posts that may have an effect on our ability to reason (Ecker et al., 2022). Misinformation spreads at alarming rates (Vosoughi et al., 2018), and an ML system should be able to quickly aid human moderators. While there is work in NLP with this goal (Markov et al., 2022; Shido et al., 2022; Ye et al., 2023), a general, practical, and open-sourced method that is effective across multiple domains remains an open challenge. Novel fake news topics or racial slurs emerge and change constantly. Retraining of ML-based systems is required to adapt this concept drift, but this is expensive, not only in terms of computation, but also in terms of the human effort needed to collect and label data.

SetFit's performance, speed, and low cost would make it ideal for effective content moderation, however, this type of text classification proves difficult for even state-of-the-art approaches. For example, detecting hate speech on Twitter (Basile et al.,

[1] Our code and data are available at https://github.com/UKPLab/lagonn.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 276–297
March 17-22, 2024 ©2024 Association for Computational Linguistics

2019), a subtask on the RAFT few-shot benchmark, appears to be the most difficult dataset; at time of writing, it is the only task where the human baseline has not been surpassed, yet SetFit is among the top ten most performant systems.[2]

Here, we propose a modification to SetFit, called Like a Good Nearest Neighbor (LAGONN). LAGONN introduces no learnable parameters and instead modifies input text by retrieving information from its nearest neighbors (NN) seen during optimization. Specifically, we append the label, distance, and text of the NNs in the training data to a new instance and encode this modified version with an ST (see Figure 1 and Table 1). By making input data appear more similar to instances seen during training, we inexpensively exploit the ST's pretrained or fine-tuned knowledge when considering a novel example. Our method can also be applied to the linear probing of an ST, requiring no expensive fine-tuning of the large embedding model. Finally, we propose a simple alteration to the SetFit training procedure, where we fine-tune the ST on a subset of the training data. This results in a more efficient and performant text classifier that can be used with LAGONN. We summarize our contributions as follows:

1. We propose LAGONN, an inexpensive modification to Sentence Transformer- or SetFit-based text classification.

2. We suggest an alternative training procedure to the standard fine-tuning of SetFit, that can be used with or without LAGONN, and results in a cheaper system with similar or improved performance to the more expensive SetFit.

3. We perform an extensive study of LAGONN, SetFit, and standard transformer fine-tuning in the context of content moderation under different label distributions, and in general and multilingual text classification settings.

2 Related Work

There is little work on using sentence embeddings as features for classification despite the pioneering work being five years old (Perone et al., 2018). STs are pretrained with the objective of maximizing the distance between semantically distinct text and minimizing the distance between text that is semantically similar in feature space. They are composed of a Siamese and triplet architecture that encodes text into dense vectors which can be used as features for ML. STs were first used to embed text for classification by Piao (2021), however, only pretrained representations were examined.

SetFit uses a contrastive learning paradigm (Koch et al., 2015; Dong et al., 2022) to optimize the ST embedding model. The ST is fine-tuned with a distance-based loss function, like cosine similarity, such that examples with different labels are separated in feature space. Input text is then encoded with the fine-tuned ST and a classifier, such as logistic regression, is trained. This approach creates a strong, few-shot text classification system, transforming the ST from a sentence encoder to a topic encoder.

Work done by Xu et al. (2021) showed that retrieving and concatenating text from training data and external sources, such as ConceptNet (Speer et al., 2017) and the Wiktionary[3] definition, can be viewed as a type of external attention that does not alter the architecture of the Transformer in question answering. Liu et al. (2022b) used PLMs and k-NN lookup to prepend examples that are similar to a GPT-3 query, aiding in prompt engineering for in-context learning. Wang et al. (2022) demonstrated that prepending and appending training data helps PLMs in summarization, language modelling, machine translation, and question answering, using BM25 as their retrieval model (Manning et al., 2008; Robertson and Zaragoza, 2009).

We alter the SetFit training procedure by using fewer examples to adapt the embedding model for many-shot learning. LAGONN decorates input text with its NN's gold label, Euclidean distance, and text from the training data to exploit both the ST's distance-based pretraining and SetFit's distance-based fine-tuning objective. Compared to retrieval-based methods, LAGONN uses the same model for both retrieval and encoding, retrieving only information from the training data for classification.

3 Like a Good Nearest Neighbor

Xu et al. (2021) formulate a type of external attention, where textual information is retrieved from multiple sources and added to text input to give the model stronger reasoning ability without altering the internal architecture. Inspired by this approach, LAGONN exploits pretrained and fine-tuned knowledge through external attention, but the

[2] https://huggingface.co/spaces/ought/raft-leaderboard (see "Tweet Eval Hate").

[3] https://www.wiktionary.org/

Training Data	Test Data
"I love this." [positive 0.0] (0)	"So good!" [?] (?)
"This is great!" [positive 0.5] (0)	"Just terrible!" [?] (?)
"I hate this." [negative 0.7] (1)	"Never again." [?] (?)
"This is awful!" [negative 1.2] (1)	"This rocks!" [?] (?)

LaGoNN Configuration	Train Modified
LABEL	"I love this. [SEP] [positive]" (0)
DISTANCE	"I love this. [SEP] [0.5]" (0)
LabDist	"I love this. [SEP] [positive 0.5]" (0)
TEXT	"I love this. [SEP] [positive 0.5] This is great!" (0)
ALL	"I love this. [SEP] [positive 0.5] This is great! [SEP] [negative 0.7] I hate this." (0)
	Test Modified
LABEL	"So good! [SEP] [positive]" (?)
DISTANCE	"So good! [SEP] [1.5]" (?)
LabDist	"So good! [SEP] [positive 1.5]
TEXT	"So good! [SEP] [positive 1.5] I love this." (?)
ALL	"So good! [SEP] [positive 1.5] I love this. [SEP] [negative 2.7] This is awful!" (?)

Table 1: Toy training and test data and different LaGoNN configurations considering the first training example. Text is in quotation marks and the integer label is in parenthesis. In brackets are the gold label or distance from the NN or both. Train and Test Modified are altered instances that are input into the final embedding model for training and inference, respectively. The input format is "*original text* [SEP] [(NN gold) (label distance)] NN *training instance text*".

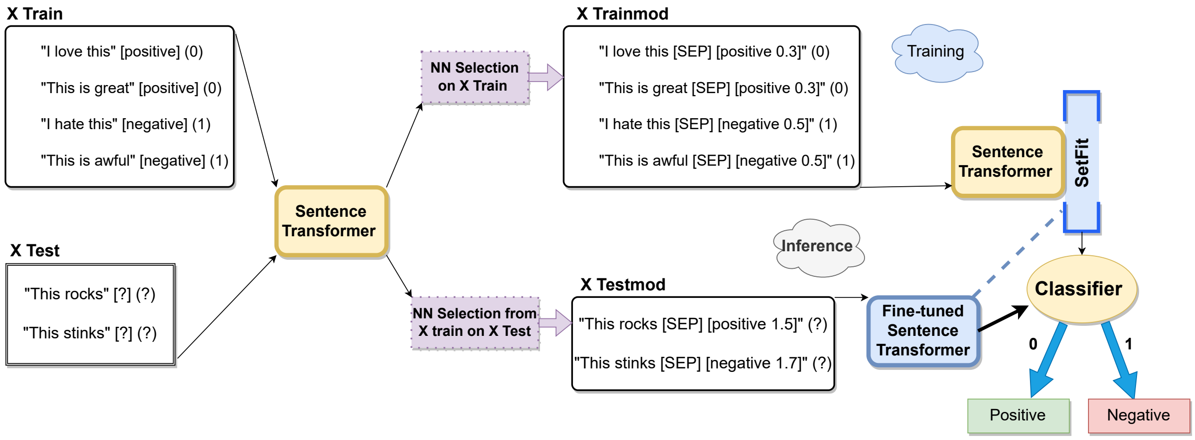
X Train
"I love this" [positive] (0)
"This is great" [positive] (0)
"I hate this" [negative] (1)
"This is awful" [negative] (1)
X Test
"This rocks" [?] (?)
"This stinks" [?] (?)
Sentence Transformer
NN Selection on X Train
NN Selection from X train on X Test
X Trainmod
"I love this [SEP] [positive 0.3]" (0)
"This is great [SEP] [positive 0.3]" (0)
"I hate this [SEP] [negative 0.5]" (1)
"This is awful [SEP] [negative 0.5]" (1)
X Testmod
"This rocks [SEP] [positive 1.5]" (?)
"This stinks [SEP] [negative 1.7]" (?)
Training
Inference
Sentence Transformer
SetFit
Fine-tuned Sentence Transformer
Classifier
0
1
Positive
Negative

Figure 1: LaGoNN LabDist uses an ST to encode training data, performs NN lookup, appends the NN's gold label and distance, and optionally SetFit to fine-tune the embedding model. We then embed this new instance and train a classifier. During inference, we use the embedding model to modify the test data with its NN's gold label and distance from the training data, compute the final representation, and call the classifier. Input text is in quotation marks, the NN's gold label and distance are in brackets, and the integer label is in parenthesis.

information we retrieve comes only from data used during optimization. We consider an embedding function, f, that encodes both training and test data, $f(X_{train})$ and $f(X_{test})$. Considering its success on realistic, few-shot data and our goal of practical content moderation, we choose an ST that can be fine-tuned with SetFit as our embedding function.

Encoding and nearest neighbors LaGoNN first uses a pretrained Sentence Transformer to embed training text in feature space, $f(X_{train})$, and NN lookup with scikit-learn (Buitinck et al., 2013) on the resulting embeddings.

Nearest neighbor information We extract text from the nearest neighbors and use it to decorate the original example. We experimented with different text that LaGoNN could use. The first configuration we consider is the gold label of the NN, which we call LABEL. We then consider the Euclidean distance of the NN, which we call DISTANCE, giving the model access to a continuous measure of similarity. We then combine these two configurations, appending both the NN's gold label and Euclidean distance, referring to this as LabDist. Next, we consider the gold label, distance, and the text of the NN, which we refer to as TEXT. Finally, we tried the same format as TEXT but for all possible labels, which we call ALL (see Table 1 and Figure 1). Information from the NN is appended to the text following a separator token to indicate this instance is composed of multiple sequences. If we consider multiple neighbors, we append the information we consider sequentially based on the Euclidean distance from the input text separated by a separator token. That is, the first NN's information is followed by "[SEP]" and the second NN's information which is then followed by "[SEP]" and the third NN's information, etc.

Training LaGoNN encodes the modified training data, optionally fine-tunes the embedding model via SetFit, and trains a classifier, $CLF(f(X_{trainmod}))$.

Inference LaGoNN uses information from the nearest neighbor in the training data to modify input text. We compute the embeddings of the test data, $f(X_{test})$, and select and extract information from the NN's training text, decorating the input instance with this information. Finally, we encode the modified data with the embedding model and call the classifier, $CLF(f(X_{testmod}))$.

Intuition The ST's pretraining and SetFit's fine-tuning objective both rely on distance, creating a feature space appropriate for distance-based algorithms, such as our NN-lookup. We hypothesize that LaGoNN's modifications make novel data appear semantically similar to their NNs in the training data, that is, more akin to an instance on which the encoder and classifier were optimized. LaGoNN's utilization of distance and clear distinctions between classes inspired our use case of content moderation, where it is realistic to have few labels, harmful or neutral, for example. However, this work demonstrates that LaGoNN is useful for general and multilingual text classification as well.

4 Experiments

We first study LaGoNN's performance on four binary and one ternary classification dataset related to the task of content moderation. Each dataset is composed of a training, validation, and test split (see Appendix A.1 for details).

We study our system by simulating growing training data over ten discrete steps sampled under four different label distributions: extreme, imbalanced, moderate, and balanced (see Table 4). On each step we add 100 examples (100 on the first, 200 on the second, etc.) from the training split sampled under one of the four ratios. On each step, we train our method with the sampled data and evaluate on the test split. Considering growing training data has two benefits: 1) We can simulate a streaming data scenario, where new data are labeled and added for training and 2) We can investigate each method's sensitivity to the number of training examples.

This experimental setup is reflective of a practical setting, where we might construct a content flagging or text classification system with a relatively small number (100) of labeled examples for training. As time goes on, however, more samples are added and we must then determine whether or not it is worth the resources to retrain our system. We sampled over five seeds, reporting the mean and standard deviation.

4.1 Baselines

We compare LaGoNN against a number of strong baselines, detailed below. We used default hyperparameters in all cases unless stated otherwise.

RoBERTa RoBERTa-base is a pretrained language model (Liu et al., 2019) that we fine-tuned with the transformers library (Wolf et al., 2020). We select two versions of RoBERTa-base: an expensive version, where we perform standard fine-tuning on each step (RoBERTa$_{full}$) and a cheaper version, where we freeze the model body after step one and update the classification head on subsequent steps (RoBERTa$_{freeze}$). We set the learning rate to $1e^{-5}$, train for a maximum of 70 epochs, and use early stopping, selecting the best model after training. We consider RoBERTa$_{full}$ an upper bound as it has the most trainable parameters and requires the most time to train of all our methods.

Linear probe We perform linear probing of a pretrained Sentence Transformer by fitting logis-

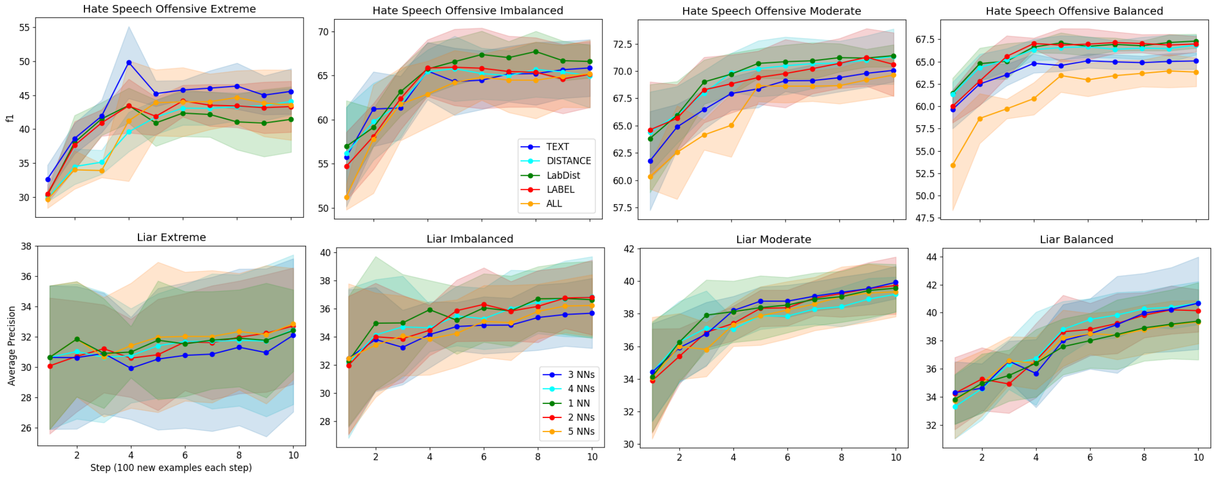

Figure 2: First row: performance for all LAGONN configurations and balance regimes for the Hate Speech Offensive dataset. Second row: LAGONN performance for one to five neighbors for all balance regimes on a collapsed version of the LIAR dataset. We use the LAGONN$_{lite}$ fine-tuning strategy (see Section 5.1).

tic regression with default hyperparameters on the training embeddings on each step. We choose this baseline because LAGONN can be applied as a modification in this scenario. We select MPNET (Song et al., 2020) as the ST, for SetFit, and for LAGONN.[4] We refer to this method as Probe.

SetFit Here, we perform standard fine-tuning with SetFit on the first step, and then on subsequent steps, freeze the embedding model and retrain only the classification head. We choose this baseline as LAGONN relies on ST/SetFit for its modifications.

***k*-nearest neighbors** Similar to the above baseline, we fine-tune the embedding model via SetFit, but swap out the classification head for a kNN classifier, where $k = 3$. We select this baseline as LAGONN also relies on an NN lookup. $k = 3$ was chosen during our development stage as it yielded the strongest performance. We refer to this method as kNN.

SetFit expensive For this baseline we perform standard fine-tuning with SetFit on each step. On the first step, this method is equivalent to SetFit. We refer to this as SetFit$_{exp}$.

LAGONN cheap This method modifies data via LAGONN before fitting logistic regression. Even without adapting the embedding model, as the training data grow, modifications made to the test data may change. Only the classification head is fit on each step. We refer to this method as LAGONN$_{cheap}$ and it is comparable to Probe.

LAGONN On the first step, we use LAGONN to modify our data and perform standard fine-tuning with SetFit. On subsequent steps, we freeze the embedding model but continue to use it to modify our data. We only fit logistic regression on later steps, referring to this method as LAGONN. It is comparable to SetFit.

LAGONN expensive Here we modify our data and fine-tune the embedding model on each step. We refer to this method as LAGONN$_{exp}$ and it is comparable to SetFit$_{exp}$. On the first step, this method is equivalent to LAGONN.

Model choices We again choose these systems to reflect different practical settings, where we might not have the resources to fine-tune our model (Probe/LAGONN$_{cheap}$), we might be able to perform limited fine-tuning (RoBERTa$_{freeze}$, SetFit, kNN, LAGONN), or we may be able to fine-tune as much as we like (RoBERTa$_{full}$, SetFit$_{exp}$, LAGONN$_{exp}$).

4.2 LAGONN configurations

We perform extensive experiments over the different LAGONN configurations. We note that while DISTANCE and LABEL show similar performance, LABDIST in general is the most performant and consistent classifier. We base this assertion on the fact that across all of our experiments, LABDIST is generally in the top three most-performant configurations and is easily the stablest, based on

[4] https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2

Method	**InsincereQs**				**AmazonCF**			
Extreme	1^{st}	5^{th}	10^{th}	Average	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$19.9_{8.4}$	$30.9_{7.9}$	$42.0_{7.4}$	$33.5_{6.7}$	$21.8_{6.6}$	$63.9_{10.2}$	$72.3_{3.0}$	$59.6_{16.8}$
SetFit$_{exp}$	$24.1_{6.3}$	$29.2_{6.7}$	$36.7_{7.3}$	$31.7_{3.4}$	$22.3_{8.8}$	$64.2_{3.3}$	$68.6_{4.6}$	$56.8_{14.9}$
LAGONN$_{exp}$	$\mathbf{30.7}_{8.9}$	$37.6_{6.1}$	$39.0_{6.1}$	$36.1_{2.3}$	$\mathbf{26.1}_{17.5}$	$\mathbf{68.4}_{4.4}$	$\mathbf{74.9}_{2.9}$	$\mathbf{63.2}_{16.7}$
RoBERTa$_{freeze}$	$19.9_{8.4}$	$34.1_{5.4}$	$37.9_{5.9}$	$32.5_{5.5}$	$21.8_{6.6}$	$41.0_{12.7}$	$51.3_{10.7}$	$40.6_{8.9}$
kNN	$6.8_{0.42}$	$15.9_{3.4}$	$16.9_{4.3}$	$14.4_{3.0}$	$10.3_{0.2}$	$15.3_{4.2}$	$18.4_{3.7}$	$15.6_{2.4}$
SetFit	$24.1_{6.3}$	$31.7_{4.9}$	$36.1_{5.4}$	$31.8_{3.6}$	$22.3_{8.8}$	$32.4_{11.5}$	$42.3_{8.8}$	$34.5_{5.9}$
LAGONN	$\mathbf{30.7}_{8.9}$	$39.3_{4.9}$	$41.2_{4.7}$	$38.4_{3.0}$	$\mathbf{26.1}_{17.5}$	$31.1_{19.4}$	$33.0_{19.1}$	$30.9_{2.3}$
Probe	$24.3_{8.4}$	$39.8_{5.6}$	$44.8_{4.2}$	$38.3_{6.2}$	$24.2_{9.0}$	$46.3_{4.4}$	$54.6_{2.0}$	$45.1_{10.3}$
LAGONN$_{cheap}$	$23.6_{7.8}$	$\mathbf{40.7}_{5.9}$	$\mathbf{45.3}_{4.4}$	$\mathbf{38.6}_{6.6}$	$20.1_{6.9}$	$38.3_{4.9}$	$47.8_{3.4}$	$38.2_{9.5}$
Balanced								
RoBERTa$_{full}$	$47.1_{4.2}$	$52.1_{3.6}$	$55.7_{2.6}$	$52.5_{2.9}$	$73.6_{2.1}$	$78.6_{3.9}$	$\mathbf{82.4}_{1.1}$	$78.9_{2.2}$
SetFit$_{exp}$	$43.5_{4.2}$	$47.1_{4.6}$	$48.5_{3.9}$	$48.0_{1.7}$	$73.8_{4.4}$	$69.8_{4.0}$	$64.1_{4.6}$	$69.6_{3.6}$
LAGONN$_{exp}$	$42.8_{5.3}$	$47.6_{2.9}$	$47.0_{1.7}$	$46.2_{2.0}$	$\mathbf{76.0}_{3.0}$	$73.4_{2.6}$	$72.3_{2.9}$	$72.5_{3.4}$
RoBERTa$_{freeze}$	$47.1_{4.2}$	$52.1_{0.4}$	$53.3_{1.7}$	$51.5_{2.1}$	$73.6_{2.1}$	$76.8_{1.6}$	$77.9_{1.0}$	$76.5_{1.3}$
kNN	$22.3_{2.3}$	$30.2_{2.3}$	$30.9_{1.8}$	$29.5_{2.5}$	$41.7_{3.4}$	$57.9_{3.3}$	$58.3_{3.3}$	$56.8_{5.1}$
SetFit	$43.5_{4.2}$	$53.8_{2.2}$	$55.5_{1.6}$	$52.8_{3.5}$	$73.8_{4.4}$	$79.2_{1.9}$	$80.1_{1.0}$	$78.6_{1.8}$
LAGONN	$42.8_{5.3}$	$54.1_{2.9}$	$56.3_{1.3}$	$53.4_{3.7}$	$\mathbf{76.0}_{3.0}$	$\mathbf{80.1}_{2.0}$	$81.4_{1.1}$	$\mathbf{79.8}_{1.4}$
Probe	$47.5_{1.6}$	$52.4_{1.7}$	$55.3_{1.1}$	$52.2_{2.5}$	$52.4_{3.4}$	$64.7_{2.5}$	$67.5_{0.4}$	$63.4_{4.4}$
LAGONN$_{cheap}$	$\mathbf{49.3}_{2.6}$	$\mathbf{54.4}_{1.4}$	$\mathbf{57.6}_{0.7}$	$\mathbf{54.2}_{2.7}$	$48.1_{3.4}$	$62.0_{2.0}$	$65.3_{0.8}$	$60.5_{5.0}$

Table 2: Average performance (average precision × 100) on Insincere Questions and Amazon Counterfactual. The first, fifth, and tenth step are followed by the average over all ten steps. The average gives insight into the overall strongest performer by aggregating all steps. We group methods with a comparable number of trainable parameters together. The extreme label distribution results are followed by balanced (see Appendix A.4 for additional results).

the standard deviation over five seeds, where DISTANCE and LABEL are less reliable and show greater oscillation. These observations are supported by Figure 2. TEXT and ALL are arguably the most interesting LAGONN configurations, but are often unstable, low-performing classifiers. In Figure 2, we provide a comparison between the different configurations on the Hate Speech Offensive dataset. As LABDIST is the most performant configuration, it is the version of our method about which we report results hereafter, and we consider it the default configuration of LAGONN. However, this is a hyperparameter that can be easily experimented with and tuned.

4.3 LAGONN k nearest neighbors

To determine how many neighbors we should consider for LAGONN, we perform thorough experiments for one to five neighbors over all datasets, LAGONN configurations, and balance regimes under the LAGONN$_{lite}$ fine-tuning strategy (see Section 5.1). We find that one to three neighbors tends to result in the strongest classifier, but this varies and is a hyperparameter that can be searched over. In Figure 2, we provide a representative example of our NN results for the LABDIST configuration for the LIAR dataset.

5 Content Moderation Results

Table 2 and Figure 5 show our results. In the cases of the extreme and imbalanced regimes, the performance of SetFit$_{exp}$ steadily increases with the number of training examples. As the label distribution shifts to the balanced regime, however, the performance quickly saturates or even degrades as the number of training examples grows. LAGONN, RoBERTa$_{full}$, and SetFit, other fine-tuned PLM classifiers, do not exhibit this behavior. LAGONN$_{exp}$, being based on SetFit$_{exp}$, exhibits a similar trend, but the performance degradation is mitigated; on the 10^{th} step of Amazon Counterfactual in Table 2 SetFit$_{exp}$'s performance decreased by 9.7, while LAGONN$_{exp}$ only fell by 3.7. Note that we only consider the first NN here.

LAGONN and LAGONN$_{exp}$ generally outperform SetFit and SetFit$_{exp}$, respectively, often resulting in a more stable model, as reflected in the

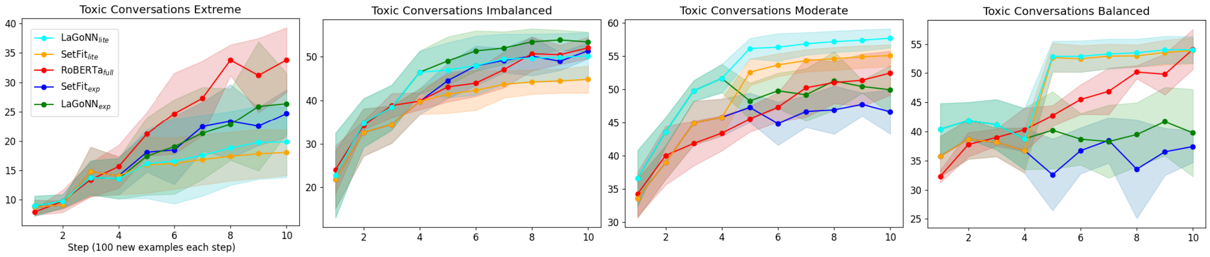

Figure 3: Average performance for all sampling regimes on Toxic Conversations. More expensive models, such as LAGONN$_{exp}$, SetFit$_{exp}$, and RoBERTa$_{full}$ perform best when the label distribution is imbalanced. As the distribution becomes more balanced, inexpensive models, such as LAGONN$_{lite}$, show similar or improved performance. The measure is average precision and we only consider one neighbor for the LAGONN-based methods (see Appendix A.4 for additional results).

standard deviation. We find that LAGONN and LAGONN$_{exp}$ exhibit stronger predictive power with fewer examples than RoBERTa$_{full}$ despite having fewer trainable parameters. On the first step of Insincere Questions under the extreme setting, LAGONN's performance is more than 10 points higher.

LAGONN$_{cheap}$ outperforms all other methods on the Insincere Questions dataset for all balance regimes, despite being the third fastest (see Table 6) and having the second fewest trainable parameters. We attribute this result to the fact that this dataset is composed of questions from Quora[5] and our ST backbone was pretrained on similar data. This intuition is supported by Probe, the cheapest method, which despite having the fewest trainable parameters, shows comparable performance.

5.1 SetFit for efficient many-shot learning

Respectively comparing SetFit to SetFit$_{exp}$ and LAGONN to LAGONN$_{exp}$ suggests that fine-tuning the ST embedding model on moderate or balanced data hurts model performance as the number of training samples grows. We therefore hypothesize that randomly sampling a subset of training data to fine-tune the encoder, freezing, embedding the remaining data, and training the classifier will result in a stronger model.

To test our hypothesis, we add two models to our experimental setup: SetFit$_{lite}$ and LAGONN$_{lite}$. SetFit$_{lite}$ and LAGONN$_{lite}$ are respectively equivalent to SetFit$_{exp}$ and LAGONN$_{exp}$, except after the fourth step (400 samples), we freeze the encoder and only retrain the classifier on subsequent steps, similar to SetFit and LAGONN.

Figures 3 and 6 show our results with these two new models. As expected, in the cases of extreme and imbalanced distributions, LAGONN$_{exp}$, SetFit$_{exp}$, and RoBERTa$_{full}$, are the strongest performers. We note very different results for both LAGONN$_{lite}$ and SetFit$_{lite}$ compared to LAGONN$_{exp}$ and SetFit$_{exp}$ on Toxic Conversations under the moderate and balanced label distributions. As their expensive counterparts start to plateau or degrade on the fourth step, these two new models dramatically increase, showing improved or comparable performance to RoBERTa$_{full}$, despite being optimized on less data; for example, LAGONN$_{lite}$ reaches an average precision of approximately 55 after being optimized on only 500 examples. RoBERTa$_{full}$ does not exhibit similar performance until the tenth step. Finally, we point out that LAGONN-based methods generally provide a performance boost for SetFit-based methods.

6 LAGONN as a General Classifier

LAGONN is effective for general text classification. Thus far, we have focused on the important topic of content moderation, but here we turn our attention to general text classification, conducting experiments on 12 additional datasets (see Appendix A.2 for details and Appendix A.6 for multilingual experiments). Our experimental setup remains largely the same, but here we restrict ourselves to the balanced sampling regime as it is nontrivial to design sampling strategies for datasets with more than three labels. We respectively compare LAGONN$_{lite}$ against SetFit$_{lite}$ and LAGONN$_{exp}$ against SetFit$_{exp}$, showing results for one to five neighbors with LAGONN.

In Figure 4, we demonstrate that LAGONN continues to stabilize and improve SetFit, regardless of the number of neighbors we consider. This is especially clear for IMDB, where in the case of LAGONN$_{lite}$ vs SetFit$_{lite}$, all versions of our

[5] https://www.quora.com/

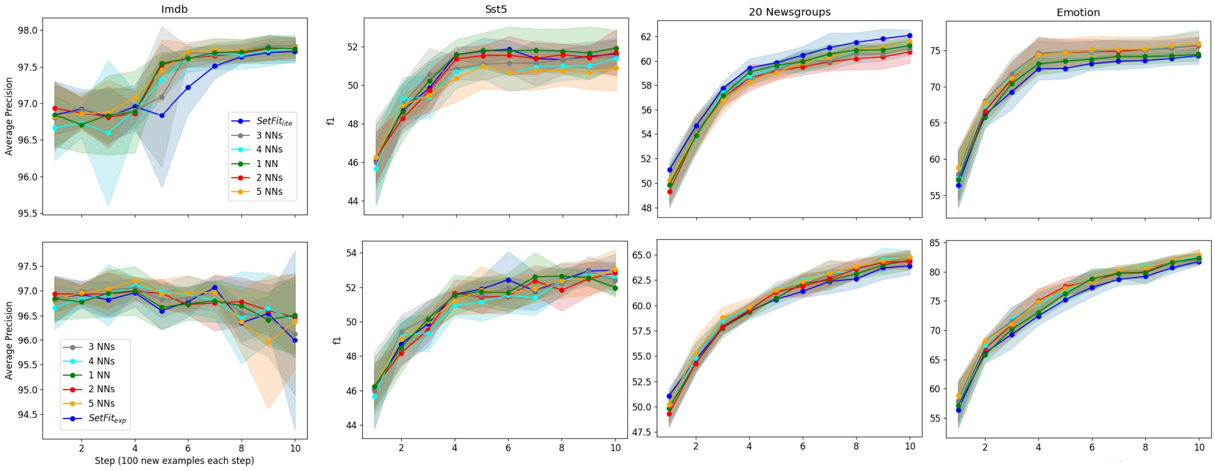

Figure 4: Average performance on four datasets in the balanced sampling regime; the measure is average precision for IMDB, macro-f1 elsewhere. First row: SetFit$_{lite}$ compared to LAGONN$_{exp}$ LABDIST with modifications for one to five neighbors. Second row: SetFit$_{exp}$ compared to LAGONN$_{exp}$. See Appendix A.5 for additional results.

method saturate to an average precision of 98 with 300 fewer training samples. If we consider SetFit$_{exp}$ vs LAGONN$_{exp}$, consistent with our analysis of other binary datasets, classifier performance begins to degrade if we continue to fine-tune the ST, but LAGONN mitigates this performance drop.

Continuing to fine-tune the embedding model is beneficial when we have many labels. For 20 Newsgroups and Emotion, which have 20 and 28 labels respectively, LAGONN$_{exp}$ is the strongest model and shows no indication of plateauing or degrading, even with 1,000 samples. We attribute this to the relatively high number of labels present in both of these datasets. Our findings related to SST-5 and our multilingual experiments (see Appendix A.6) support this; in intermediate cases when we have five labels, all models saturate quickly and there are minimal gains with continued fine-tuning.

7 Discussion

Flagging potentially dangerous text presents a challenge even for state-of-the-art approaches. The content moderation datasets we consider proved more difficult than our general text classification datasets for all models, despite typically having fewer labels. It is imperative that we develop reliable and practical text classifiers for content moderation, such that we can inexpensively re-tune them for novel forms of hate speech, toxicity, and fake news.

Our results suggest that LAGONN$_{exp}$, a relatively expensive technique, can detect harmful content when dealing with imbalanced label distributions, as is common with realistic datasets. This is intuitive from the perspective that less common instances are more difficult to learn and require more effort. An exception would be our examination of Insincere Questions, where LAGONN$_{cheap}$ excelled in the extreme and balanced settings. This demonstrates that if we choose our PLM with care for related downstream tasks, LAGONN can inexpensively extract pretrained knowledge and improve performance without the need for costly fine-tuning. Indeed, considering the performance of SetFit suggests that, in this case, fine-tuning hurts performance and we actually overfit. However, even here, our proposed modifications with LAGONN increase model robustness and lessen the effects of overfitting.

Fine-tuning with SetFit hurts performance on more balanced datasets that are not few-shot. We have observed that SetFit should not be applied "out of the box" to balanced, non-few-shot data. This can be detrimental to performance, directly affecting our own approach. However, LAGONN can stabilize SetFit's predictions and reduce its performance drop in many cases. Figures 5, 3, and 4 show that when the label distribution is moderate or balanced (see Table 4), SetFit$_{exp}$ plateaus, yet cheaper systems, such as LAGONN, continue to learn. This is likely due to SetFit's fine-tuning objective, which optimizes an ST using cosine similarity loss to separate examples belonging to different labels in feature space, assuming independence between labels. This may be too strong an assumption as we fine-tune with more data, which is counter-intuitive for data-hungry transformers;

RoBERTa$_{full}$, optimized with cross-entropy loss, showed improved performance as we added training data data.

For balanced data, it is sufficient to fine-tune the Sentence Transformer via SetFit with 50 to 100 examples per label, while 150 to 200 instances appear to be sufficient when the training data are moderately balanced. The encoder can then be frozen and all available data embedded to train a classifier. This is more performant and efficient than full-model fine-tuning. LAGONN is applicable to this case, inexpensively boosting and stabilizing SetFit's performance. All models fine-tuned on Hate Speech Offensive exhibited similar, upward-trending learning curves, but we note the speed of LAGONN relative to RoBERTa$_{full}$ or SetFit$_{exp}$ (see Figure 3 and Table 6).

8 Conclusion

We have proposed LAGONN, an inexpensive modification to SetFit. LAGONN improves SetFit's performance by modifying text with the nearest neighbors in the training data. To demonstrate the merit of LAGONN, we examined text classification systems for content moderation with different label distributions and for general and multilingual classification. We studied 17 datasets with growing training data. When the training labels are imbalanced, expensive systems, such as LAGONN$_{exp}$ are performant. LAGONN$_{exp}$ also excels on balanced datasets with many labels. However, when the labels are binary or ternary, typical for content moderation, and the distribution is balanced, fine-tuning with SetFit yields minimal gains. We therefore proposed an alternative but strong training procedure. LAGONN is a practical method for detecting harmful content and text classification.

9 Acknowledgments

We would like to thank Derek Hommel and Nils Reimers for sharing inspiring discussions with us. We would also like to extend our gratitude to Tom Aarsen, Max Glockner, Yongxin Huang, Timour Igamberdiev, Sukannya Purkayastha, and Kexin Wang for their invaluable feedback on an early draft of our manuscript. This work was funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Science and the Arts (HMWK) within the projects "The Third Wave of Artificial Intelligence - 3AI", hessian.AI, and within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

10 Limitations

In the current work, we have only considered text data, but social media content can of course consist of text, images, and videos. As LAGONN depends only on an embedding model, an obvious extension to our approach would be examining the modifications we suggest, but on multimodal data. This is an interesting direction that we leave for future research. We did not study our method when there are fewer than 100 training examples, and investigating LAGONN in a few-shot learning setting is fascinating topic for future study. Finally, we note that our system could be misused to detect undesirable content that is not necessarily harmful. For example, a social media website could detect and silence users who complain about the platform. This is not our intended use case, but could result from any classifier, and potential misuse is an unfortunate drawback of all technology.

11 Ethics Statement

It is our sincere goal that our work contributes to the social good in multiple ways. We first hope to have furthered research on text classification that can be feasibly applied to combat undesirable content, such as misinformation, on the Internet, which could potentially cause someone harm. To this end, we have tried to describe our approach as accurately as possible and released our code and data, such that our work is transparent and can be easily reproduced and expanded upon. We hope that we have also created a useful but efficient system which reduces the need to expend energy in the form expensive computation. For example, LAGONN does not rely on billion-parameter language models that demand thousand-dollar GPUs to use. LAGONN makes use of GPUs no more than SetFit, despite being more computationally expensive. We have additionally proposed a simple method to make SetFit, an already relatively inexpensive method, even more efficient.

References

Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C. Jess Riedel, Emmie Hine, Carolyn Ashurst, Paul Sedille, Alexis Carlier, Michael Noetel, and Andreas Stuhlmüller. 2021. RAFT: A real-world few-shot text classification benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

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.

Bo Dong, Yiyi Wang, Hanbo Sun, Yunji Wang, Alireza Hashemi, and Zheng Du. 2022. CML: A contrastive meta learning method to estimate human label confidence scores and reduce data collection cost. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 35–43, Dublin, Ireland. Association for Computational Linguistics.

Ullrich K. H. Ecker, Stephan Lewandowsky, John Cook, Philipp Schmid, Lisa K. Fazio, Nadia Brashier, Panayiota Kendeou, Emily K. Vraga, and Michelle A. Amazeen. 2022. The psychological drivers of misinformation belief and its resistance to correction. *Nature Reviews Psychology*, 1(1):13–29.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. Prompt-free and efficient few-shot learning with language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3638–3652, Dublin, Ireland. Association for Computational Linguistics.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, page 0. Lille.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv:2205.05638*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022b. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, USA.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2022. A holistic approach to undesired content detection. *arXiv preprint arXiv:2208.03274*.

Tom Mitchell. 1999. Twenty Newsgroups. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5C323.

James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. I wish I would have loved this one, but I didn't – a multilingual dataset for counterfactual detection in product review. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7092–7108, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Christian S. Perone, Roberto Pereira Silveira, and Thomas S. Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Guangyuan Piao. 2021. Scholarly text classification with sentence bert and entity embeddings. In *Trends and Applications in Knowledge Discovery and Data Mining*, pages 79–87, Cham. Springer International Publishing.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sarah T. Roberts. 2017. *Content Moderation*, pages 1–4. Springer International Publishing, Cham.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2022. True few-shot learning with Prompts—A real-world perspective. *Transactions of the Association for Computational Linguistics*, 10:716–731.

Yusuke Shido, Hsien-Chi Liu, and Keisuke Umezawa. 2022. Textual content moderation in C2C marketplace. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 58–62, Dublin, Ireland. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3170–3179, Dublin, Ireland. Association for Computational Linguistics.

William Yang Wang. 2017. “Liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

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.

Yichong Xu, Chenguang Zhu, Shuohang Wang, Siqi Sun, Hao Cheng, Xiaodong Liu, Jianfeng Gao, Pengcheng He, Michael Zeng, and Xuedong Huang. 2021. Human parity on commonsenseqa: Augmenting self-attention with external attention. *arXiv preprint arXiv:2112.03254*, abs/2112.03254.

Meng Ye, Karan Sikka, Katherine Atwell, Sabit Hassan, Ajay Divakaran, and Malihe Alikhani. 2023. Multilingual content moderation: A case study on reddit. *arXiv preprint arXiv:2302.09618*.

A Appendix

A.1 Content moderation data and balance regimes

In this Appendix section, we provide a background on the datasets we studied in our experiments and summarize the label distribution (see Table 3) of our content moderation datasets and the different sampling regimes (see Table 4) we studied in our content moderation experiments. LIAR was created from Politifact[6] for fake news detection and is composed of the data fields *context*, *speaker*, and *statement*, which are labeled with varying levels of truthfulness (Wang, 2017). We used a collapsed version of this dataset where a statement can only be true or false. We did not use *speaker*, but did use *context* and *statement*, separated by a separator token. Quora Insincere Questions[7] is composed of neutral and toxic questions, where the author is not asking in good faith. Hate Speech Offensive[8] has three labels and is composed of tweets that can contain either neutral text, offensive language, or hate speech (Davidson et al., 2017).[9] Amazon Counterfactual[10] contains sentences from product reviews, and the labels can be "factual" or "counterfactual" (O’Neill et al., 2021). "Counterfactual" indicates that the customer said something that cannot be true. Finally, Toxic Conversations[11] is a dataset of comments where the author wrote with unintended bias[12] (see Table 3).

Dataset (and Detection Task)	Number of Labels
LIAR (Fake News)	2
Insincere Questions (Toxicity)	2
Hate Speech Offensive	3
Amazon Counterfactual (English)	2
Toxic Conversations	2

Table 3: Summary of content moderation datasets and number of labels. We provide the type of task in parenthesis in unclear cases.

Regime	Binary	Ternary
Extreme	0: 98% 1: 2%	0: 95%, 1: 2%, 2: 3%
Imbalanced	0: 90% 1: 10%	0: 80%, 1: 5%, 2: 15%
Moderate	0: 75% 1: 25%	0: 65%, 1: 10%, 2: 25%
Balanced	0: 50% 1: 50%	0: 33%, 1: 33%, 2: 33%

Table 4: Label distributions for sampling training data. 0 represents neutral while 1 and 2 represent different types of undesirable text.

A.2 General text classification data

In this Appendix section, we provide additional information on the datasets we examined in our general text classification experiments. The Internet Movie Database (IMDB) dataset (Maas et al., 2011) is composed of movie reviews that are classified as either positive or negative.[13] Student Question Categories contains questions from qualifying ex-

[6] https://www.politifact.com/

[7] https://www.kaggle.com/c/quora-insincere-questions-classification

[8] https://huggingface.co/datasets/hate_speech_offensive

[9] For Hate Speech Offensive, 0 and 2 denote undesirable text and 1 denotes neither.

[10] https://huggingface.co/datasets/SetFit/amazon_counterfactual_en

[11] https://huggingface.co/datasets/SetFit/toxic_conversations

[12] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

[13] https://huggingface.co/datasets/SetFit/imdb

aminations in India,[14] where the label is the subject the question appeared in and can be from Physics, Chemistry, Biology, or Mathematics.[15] SST5 is an alternative version of the Stanford Sentiment Treebank (Socher et al., 2013) that has five labels, ranging from very positive to very negative.[16] We also include the original version of LIAR, which has six labels of varying levels of truthfulness.[17] We also used 20 Newsgroups[18] (Mitchell, 1999) which contains newspaper articles labeled with the topic they cover.[19] And finally, we ran experiments on GoEmotions (Demszky et al., 2020), a dataset of Reddit comments labeled with 28 classes based on the emotional charge of the post.[20]

The evaluation measure was average precision in the case of IMDB, macro F1 elsewhere. In cases where the a validation split was not available, we created one by sampling 30% of the test split. Please see Table 5 for a summary regarding the datasets and label information.

Dataset (and Detection Task)	**Number of Labels**
IMDB (Sentiment Analysis)	2
Student Questions (Question Type)	4
SST5 (Sentiment Analysis)	5
LIAR (Fake News)	6
20 Newsgroups (Topic)	20
GoEmotions (Emotion)	28

Table 5: Summary of datasets and number of labels used in the general text classification experiments. We provide the type of task in parenthesis in unclear cases.

A.3 LaGoNN's computational expense

In this Appendix section we discuss and provide results for LaGoNN's computation time. LaGoNN is more computationally expensive than Sentence Transformer- or SetFit-based text classification. LaGoNN introduces additional inference with the encoder, NN-lookup, and string modification. As the computational complexity of transformers increases with sequence length (Vaswani et al., 2017), additional expense is created when LaGoNN appends textual information before inference with the ST. In Table 6, we provide a speed comparison of comparable methods computed on the same hardware.[21] On average, LaGoNN introduced 24.2 additional seconds of computation compared to its relative counterpart.

Method	**Time in seconds**
Probe	22.9
LaGoNN$_{cheap}$	44.2
SetFit	42.9
LaGoNN	63.4
SetFit$_{exp}$	207.3
LaGoNN$_{exp}$	238.0
RoBERTa$_{full}$	446.9

Table 6: Speed comparison between LaGoNN LabDist with one neighbor and comparable methods. Time includes training on $1,000$ examples and inference on $51,000$ examples.

[14] https://www.kaggle.com/datasets/mrutyunjaybiswal/iitjee-neet-aims-students-questions-data

[15] https://huggingface.co/datasets/SetFit/student-question-categories

[16] https://huggingface.co/datasets/SetFit/sst5

[17] https://huggingface.co/datasets/LIAR

[18] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html#the-20-newsgroups-text-dataset

[19] https://huggingface.co/datasets/SetFit/20_newsgroups

[20] https://huggingface.co/datasets/SetFit/go_emotions

[21] We used a 40 GB NVIDIA A100 Tensor Core GPU.

A.4 Additional results: content moderation

Here, we provide additional results from content moderation experiments that, due to space limitations, could not be included in the main text. We note that a version of LaGoNN outperforms or has the same performance of all methods, including our upper bound RoBERTa$_{full}$, on 60% of all displayed results, and is the best performer relative to Sentence Transformer-based methods on 65%. This excludes LaGoNN$_{cheap}$. This method showed strong performance on the Insincere Questions dataset, but hurts performance in other cases.

In cases when SetFit-based methods do outperform our system, the performances are comparable, usually within one point, yet they can be quite different when LaGoNN-based methods are the strongest. Below, we report the mean average precision $\times 100$ for all methods over five seeds with the standard deviation, except in the case of Hate Speech Offensive, where the evaluation measure is the macro-F1. Each table shows the results for a given dataset and a given label-balance distribution on the first, fifth, and tenth step followed by the average for all ten steps. In the table caption we provide a summary/interpretation of the results for a given setting. LIAR appears to be the most difficult dataset for all methods. This is expected because it likely does not include enough context to determine the truth of a statement.

Method	**Insincere Questions**			
Extreme	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$19.9_{8.4}$	$30.9_{7.9}$	$42.0_{7.4}$	$33.5_{6.7}$
SetFit$_{exp}$	$24.1_{6.3}$	$29.2_{6.7}$	$36.7_{7.3}$	$31.7_{3.4}$
LaGoNN$_{exp}$	$\mathbf{30.7}_{8.9}$	$37.6_{6.1}$	$39.0_{6.1}$	$36.1_{2.3}$
SetFit$_{lite}$	$24.1_{6.3}$	$38.1_{6.3}$	$41.1_{6.5}$	$35.6_{5.5}$
LaGoNN$_{lite}$	$\mathbf{30.7}_{8.9}$	$\mathbf{41.8}_{8.3}$	$\mathbf{43.4}_{8.5}$	$\mathbf{39.3}_{4.4}$
RoBERTa$_{freeze}$	$19.9_{8.4}$	$34.1_{5.4}$	$37.9_{5.2}$	$32.5_{5.4}$
kNN	$6.8_{0.4}$	$15.9_{3.4}$	$16.9_{4.3}$	$14.4_{3.0}$
SetFit	$24.1_{6.3}$	$31.7_{4.9}$	$36.1_{5.4}$	$31.8_{3.6}$
LaGoNN	$\mathbf{30.7}_{8.9}$	$39.3_{4.9}$	$41.2_{4.7}$	$38.4_{3.0}$
Probe	$24.3_{8.4}$	$39.8_{5.6}$	$44.8_{4.2}$	$38.3_{6.2}$
LaGoNN$_{cheap}$	$23.6_{7.8}$	$40.7_{5.9}$	$45.3_{4.4}$	$38.6_{6.6}$

Table 7: LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out as the strongest models, but LaGoNN$_{lite}$ remains the most performant by the 10^{th} step. It is also the overall strongest performer based on the average. We note the strength of LaGoNN$_{cheap}$ relative to far more expensive methods.

Method	**Insincere Questions**			
Imbalanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$39.8_{5.5}$	$53.1_{4.6}$	$55.7_{1.2}$	$50.6_{4.4}$
SetFit$_{exp}$	$43.7_{2.7}$	$52.2_{1.9}$	$53.8_{0.9}$	$51.4_{2.9}$
LaGoNN$_{exp}$	$\mathbf{44.5}_{4.5}$	$52.7_{2.4}$	$55.4_{2.0}$	$51.8_{3.0}$
SetFit$_{lite}$	$43.7_{2.7}$	$52.9_{2.6}$	$55.8_{1.8}$	$52.2_{3.4}$
LaGoNN$_{lite}$	$\mathbf{44.5}_{4.5}$	$\mathbf{53.5}_{2.7}$	$\mathbf{55.9}_{2.4}$	$\mathbf{52.6}_{3.5}$
RoBERTa$_{freeze}$	$39.8_{5.5}$	$44.1_{3.6}$	$46.3_{2.4}$	$44.0_{2.0}$
kNN	$23.9_{2.2}$	$30.3_{3.0}$	$31.6_{2.4}$	$30.0_{2.1}$
SetFit	$43.7_{2.7}$	$47.6_{1.6}$	$50.1_{2.1}$	$47.6_{1.8}$
LaGoNN	$\mathbf{44.5}_{4.5}$	$48.1_{2.2}$	$50.3_{1.7}$	$48.1_{1.9}$
Probe	$40.4_{4.2}$	$49.4_{2.3}$	$52.3_{1.7}$	$49.0_{3.3}$
LaGoNN$_{cheap}$	$40.8_{4.3}$	$51.1_{2.4}$	$54.5_{1.4}$	$50.4_{4.0}$

Table 8: LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out as the strongest models, but LaGoNN$_{lite}$ remains the most performant by the 10^{th} step. It is also the overall strongest performer based on the average. We note the strength of LaGoNN$_{cheap}$ relative to far more expensive methods.

Method	**Insincere Questions**			
Moderate	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$48.1_{2.3}$	$54.7_{1.9}$	$57.5_{1.5}$	$53.9_{2.9}$
SetFit$_{exp}$	$48.9_{1.7}$	$53.9_{0.7}$	$54.2_{1.5}$	$52.3_{1.6}$
LaGoNN$_{exp}$	$\mathbf{49.8}_{1.6}$	$52.2_{1.9}$	$53.2_{3.3}$	$52.0_{1.4}$
SetFit$_{lite}$	$48.9_{1.7}$	$\mathbf{56.5}_{1.4}$	$\mathbf{58.7}_{0.6}$	$\mathbf{55.0}_{3.5}$
LaGoNN$_{lite}$	$\mathbf{49.8}_{1.6}$	$56.1_{2.8}$	$58.3_{1.5}$	$54.6_{3.5}$
RoBERTa$_{freeze}$	$48.1_{2.3}$	$50.2_{2.2}$	$52.0_{1.4}$	$50.2_{1.4}$
kNN	$28.0_{2.4}$	$33.9_{2.8}$	$33.6_{2.0}$	$33.5_{1.9}$
SetFit	$48.9_{1.7}$	$53.6_{1.9}$	$55.8_{1.7}$	$53.3_{2.2}$
LaGoNN	$\mathbf{49.8}_{1.6}$	$54.4_{1.3}$	$56.9_{0.5}$	$54.2_{2.2}$
Probe	$45.7_{2.1}$	$52.3_{1.8}$	$54.4_{1.1}$	$51.4_{2.5}$
LaGoNN$_{cheap}$	$45.7_{2.2}$	$54.4_{1.6}$	$56.4_{0.6}$	$53.2_{3.2}$

Table 9: LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out as the strongest models, but SetFit$_{lite}$ overtakes the other methods by the 5^{th} step and is the strongest performer based on the average. We note the strength of LaGoNN$_{cheap}$ relative to far more expensive methods.

Method	**Insincere Questions**			
Balanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$47.1_{4.2}$	$52.1_{3.6}$	$55.7_{2.6}$	$52.5_{2.9}$
SetFit$_{exp}$	$43.5_{4.2}$	$47.1_{4.6}$	$48.5_{3.9}$	$48.0_{1.7}$
LaGoNN$_{exp}$	$42.8_{5.3}$	$47.6_{2.9}$	$47.0_{1.7}$	$46.2_{2.0}$
SetFit$_{lite}$	$43.5_{4.2}$	$\mathbf{54.6}_{2.4}$	$\mathbf{59.6}_{0.9}$	$53.6_{5.8}$
LaGoNN$_{lite}$	$42.8_{5.3}$	$53.5_{3.7}$	$58.6_{2.5}$	$52.2_{6.4}$
RoBERTa$_{freeze}$	$47.1_{4.2}$	$52.1_{0.4}$	$53.3_{1.1}$	$51.5_{2.1}$
kNN	$22.3_{2.3}$	$30.2_{2.3}$	$30.9_{1.8}$	$29.5_{2.5}$
SetFit	$43.5_{4.2}$	$53.8_{2.2}$	$55.5_{1.6}$	$52.8_{3.5}$
LaGoNN	$42.8_{5.3}$	$54.1_{2.9}$	$56.3_{1.3}$	$53.4_{3.7}$
Probe	$47.5_{1.6}$	$52.4_{1.7}$	$55.3_{1.1}$	$52.2_{2.5}$
LaGoNN$_{cheap}$	$\mathbf{49.3}_{2.6}$	$54.4_{1.4}$	$57.6_{0.7}$	$\mathbf{54.2}_{2.7}$

Table 10: LaGoNN$_{cheap}$, starts out as the strongest model, but SetFit$_{lite}$ overtakes the other methods on the 5^{th} and 10^{th} step. Overall LaGoNN$_{cheap}$ is the strongest model despite being one of the least expensive.

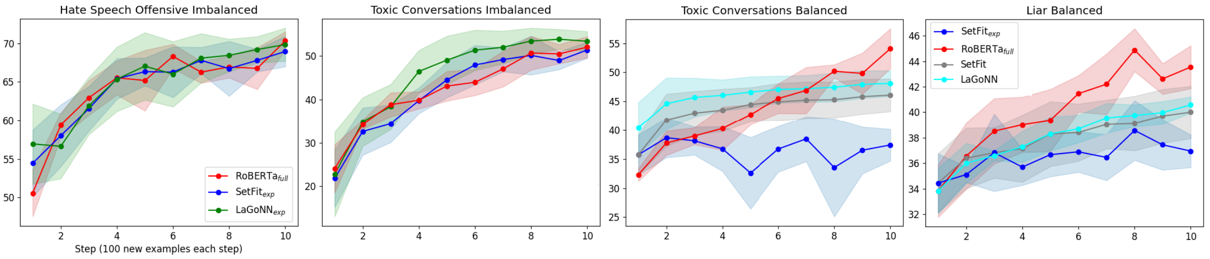

Figure 5: Average performance in the imbalanced and balanced regimes relative to comparable methods. We include RoBERTa$_{full}$ results for reference. The measure is macro-F1 for Hate Speech Offensive, average precision elsewhere.

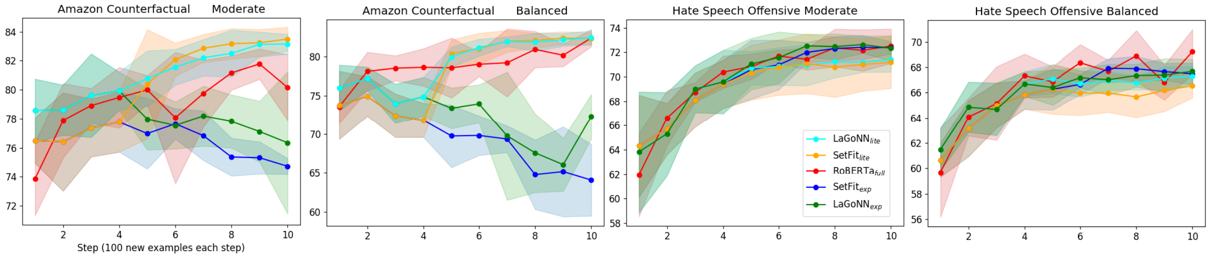

Figure 6: Average performance for all the moderate and balanced sampling regimes on Amazon Counterfactual and Hate Speech Offensive. More expensive models, such as LaGoNN$_{exp}$, SetFit$_{exp}$, and RoBERTa$_{full}$ perform best when the label distribution is imbalanced. As the distribution becomes more balanced, inexpensive models, such as LaGoNN$_{lite}$, show similar or improved performance. The measure is average precision for Amazon Counterfactual and the macro F1 for Hate Speech Offensive. We only consider one neighbor for the LaGoNN-based methods.

Method		**Amazon Counterfactual**		
Extreme	1st	5th	10th	Average
RoBERTa$_{full}$	21.8$_{6.6}$	63.9$_{10.2}$	72.3$_{3.0}$	59.6$_{16.8}$
SetFit$_{exp}$	22.3$_{8.8}$	64.2$_{3.3}$	68.6$_{4.6}$	56.8$_{14.9}$
LaGoNN$_{exp}$	**26.1**$_{17.5}$	**68.4**$_{4.4}$	**74.9**$_{2.9}$	**63.2**$_{16.7}$
SetFit$_{lite}$	22.3$_{8.8}$	62.4$_{5.1}$	67.5$_{5.2}$	56.5$_{14.7}$
LaGoNN$_{lite}$	**26.1**$_{17.5}$	68.3$_{4.3}$	68.9$_{4.3}$	60.6$_{15.1}$
RoBERTa$_{freeze}$	21.8$_{6.6}$	41.0$_{12.7}$	51.3$_{10.7}$	40.6$_{8.9}$
kNN	10.3$_{0.2}$	15.3$_{4.2}$	18.4$_{3.7}$	15.6$_{2.4}$
SetFit	22.3$_{8.8}$	32.4$_{11.5}$	42.3$_{8.8}$	34.5$_{5.9}$
LaGoNN	**26.1**$_{17.5}$	31.1$_{19.4}$	33.0$_{19.1}$	30.9$_{2.3}$
Probe	24.2$_{9.0}$	46.3$_{4.4}$	54.6$_{2.0}$	45.1$_{10.3}$
LaGoNN$_{cheap}$	20.1$_{6.9}$	38.3$_{4.9}$	47.8$_{3.4}$	38.2$_{9.5}$

Table 11: LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ are the most performant models on the first step, but only LaGoNN$_{exp}$ remains the most performant on subsequent steps, also being the strongest overall method based on the average over all steps.

Method		**Amazon Counterfactual**		
Imbalanced	1st	5th	10th	Average
RoBERTa$_{full}$	68.2$_{4.5}$	**81.0**$_{1.7}$	**82.2**$_{1.0}$	79.2$_{3.9}$
SetFit$_{exp}$	72.0$_{2.1}$	78.4$_{2.8}$	78.8$_{1.2}$	78.0$_{2.1}$
LaGoNN$_{exp}$	**74.3**$_{3.8}$	80.1$_{1.4}$	79.0$_{1.6}$	79.5$_{1.9}$
SetFit$_{lite}$	72.0$_{2.1}$	79.1$_{1.4}$	81.6$_{1.3}$	79.1$_{2.7}$
LaGoNN$_{lite}$	**74.3**$_{3.8}$	79.2$_{1.7}$	81.9$_{1.1}$	**80.2**$_{2.2}$
RoBERTa$_{freeze}$	68.2$_{4.5}$	75.0$_{2.2}$	77.0$_{2.4}$	74.2$_{2.6}$
kNN	51.0$_{4.1}$	60.0$_{3.1}$	61.3$_{2.1}$	59.7$_{3.0}$
SetFit	72.0$_{2.1}$	74.4$_{2.3}$	76.7$_{1.8}$	74.8$_{1.4}$
LaGoNN	**74.3**$_{3.8}$	76.1$_{3.6}$	77.3$_{3.2}$	76.1$_{1.0}$
Probe	46.6$_{2.8}$	60.3$_{1.4}$	64.2$_{1.2}$	59.2$_{5.2}$
LaGoNN$_{cheap}$	38.2$_{3.2}$	55.3$_{1.8}$	61.0$_{1.2}$	54.4$_{6.7}$

Table 12: On the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest but LaGoNN$_{lite}$ performs slightly worse than RoBERTa$_{full}$ on the 5th and 10th step. However, LaGoNN$_{lite}$ is the best overall method based on the average.

Method	**Amazon Counterfactual**			
Moderate	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$73.9_{2.5}$	$80.0_{1.0}$	$80.1_{2.3}$	$79.1_{2.1}$
SetFit$_{exp}$	$76.5_{1.6}$	$77.0_{2.4}$	$74.7_{0.5}$	$76.5_{1.0}$
LaGoNN$_{exp}$	$\mathbf{78.6}_{2.2}$	$78.0_{2.1}$	$76.3_{4.9}$	$78.2_{1.0}$
SetFit$_{lite}$	$76.5_{1.6}$	$80.4_{3.8}$	$\mathbf{83.5}_{0.8}$	$80.3_{2.8}$
LaGoNN$_{lite}$	$\mathbf{78.6}_{2.2}$	$80.8_{1.9}$	$83.1_{0.7}$	$\mathbf{81.0}_{1.7}$
RoBERTa$_{freeze}$	$73.9_{2.5}$	$76.6_{1.4}$	$78.5_{0.7}$	$76.4_{1.7}$
kNN	$54.5_{3.1}$	$64.2_{1.9}$	$66.6_{1.3}$	$64.7_{3.5}$
SetFit	$76.5_{1.6}$	$80.6_{0.5}$	$81.2_{0.3}$	$80.0_{1.4}$
LaGoNN	$\mathbf{78.6}_{2.2}$	$\mathbf{81.2}_{1.4}$	$81.6_{1.1}$	$80.8_{0.9}$
Probe	$52.3_{2.0}$	$64.1_{1.8}$	$67.2_{1.4}$	$63.1_{4.3}$
LaGoNN$_{cheap}$	$47.3_{3.4}$	$60.7_{1.5}$	$65.2_{1.4}$	$59.5_{5.2}$

Table 13: On the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest. On the 5^{th} step, LaGoNN is the most performant method while on the 10^{th} step it is SetFit$_{lite}$. However, LaGoNN$_{lite}$ is the best overall method based on the average.

Method	**Amazon Counterfactual**			
Balanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$73.6_{2.1}$	$78.6_{3.9}$	$82.4_{1.1}$	$78.9_{2.2}$
SetFit$_{exp}$	$73.8_{4.4}$	$69.8_{4.0}$	$64.1_{4.6}$	$69.6_{3.6}$
LaGoNN$_{exp}$	$\mathbf{76.0}_{3.0}$	$73.4_{2.6}$	$72.3_{2.9}$	$72.5_{3.4}$
SetFit$_{lite}$	$73.8_{4.4}$	$\mathbf{80.4}_{1.8}$	$82.4_{0.8}$	$78.3_{4.3}$
LaGoNN$_{lite}$	$\mathbf{76.0}_{3.0}$	$80.0_{1.3}$	$\mathbf{82.5}_{0.9}$	$79.2_{3.2}$
RoBERTa$_{freeze}$	$73.6_{2.1}$	$76.8_{1.6}$	$77.9_{1.0}$	$76.5_{1.3}$
kNN	$41.7_{3.4}$	$57.9_{3.3}$	$58.3_{3.3}$	$56.8_{5.1}$
SetFit	$73.8_{4.4}$	$79.2_{1.9}$	$80.1_{1.0}$	$78.6_{1.8}$
LaGoNN	$\mathbf{76.0}_{3.0}$	$80.1_{2.0}$	$81.4_{1.1}$	$\mathbf{79.8}_{1.4}$
Probe	$52.4_{3.4}$	$64.7_{2.5}$	$67.5_{0.4}$	$63.4_{4.4}$
LaGoNN$_{cheap}$	$48.1_{3.4}$	$62.0_{2.0}$	$65.3_{0.8}$	$60.5_{5.0}$

Table 14: On the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest. On the 5^{th} step, SetFit$_{lite}$ pulls ahead slightly, yet on the 10^{th} step LaGoNN$_{lite}$ is the best performer. Overall, LaGoNN is the best method based on the average.

Method	**Toxic Conversations**			
Extreme	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$7.9_{0.5}$	$21.2_{3.7}$	$33.8_{5.5}$	$21.9_{9.3}$
SetFit$_{exp}$	$8.8_{1.2}$	$18.1_{3.4}$	$24.7_{4.1}$	$17.6_{5.5}$
LaGoNN$_{exp}$	$8.9_{1.7}$	$17.4_{6.6}$	$26.4_{5.2}$	$17.9_{6.0}$
SetFit$_{lite}$	$8.8_{1.2}$	$15.9_{4.8}$	$18.0_{3.9}$	$14.9_{3.2}$
LaGoNN$_{lite}$	$8.9_{1.7}$	$16.1_{5.9}$	$19.8_{6.0}$	$15.5_{3.7}$
RoBERTa$_{freeze}$	$7.9_{0.5}$	$12.8_{2.4}$	$19.1_{3.2}$	$13.5_{3.5}$
kNN	$7.9_{0.0}$	$8.7_{0.4}$	$8.7_{0.2}$	$8.5_{0.3}$
SetFit	$8.8_{1.2}$	$13.1_{2.5}$	$16.3_{3.0}$	$13.0_{2.6}$
LaGoNN	$8.9_{1.7}$	$13.8_{3.9}$	$17.1_{4.8}$	$13.4_{2.6}$
Probe	$\mathbf{13.1}_{2.8}$	$\mathbf{24.6}_{2.6}$	$\mathbf{30.1}_{2.1}$	$\mathbf{23.9}_{5.6}$
LaGoNN$_{cheap}$	$11.3_{2.2}$	$21.7_{2.7}$	$27.4_{2.3}$	$21.3_{5.3}$

Table 15: Probe is most performant method on all steps and the overall strongest performer. We note, however, that LaGoNN-based methods tend to outperform their SetFit-based counterparts.

Method	**Toxic Conversations**			
Imbalanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$\mathbf{24.1}_{5.6}$	$43.1_{3.4}$	$52.1_{2.5}$	$42.4_{8.2}$
SetFit$_{exp}$	$21.8_{6.6}$	$44.5_{4.1}$	$51.4_{1.9}$	$42.1_{9.3}$
LaGoNN$_{exp}$	$22.7_{9.8}$	$\mathbf{49.1}_{5.6}$	$\mathbf{53.4}_{2.3}$	$\mathbf{45.6}_{9.8}$
SetFit$_{lite}$	$21.8_{6.6}$	$41.4_{4.4}$	$44.8_{3.1}$	$39.0_{7.0}$
LaGoNN$_{lite}$	$22.7_{9.8}$	$47.0_{6.3}$	$50.2_{5.4}$	$43.7_{8.6}$
RoBERTa$_{freeze}$	$\mathbf{24.1}_{5.6}$	$31.2_{4.4}$	$34.0_{4.0}$	$30.5_{3.1}$
kNN	$11.5_{2.5}$	$14.7_{4.0}$	$15.3_{3.2}$	$14.6_{1.1}$
SetFit	$21.8_{6.6}$	$26.7_{5.3}$	$30.2_{4.0}$	$26.6_{2.7}$
LaGoNN	$22.7_{9.8}$	$27.6_{8.9}$	$30.3_{8.7}$	$27.4_{2.4}$
Probe	$23.3_{2.7}$	$33.0_{2.8}$	$37.1_{1.8}$	$32.5_{4.2}$
LaGoNN$_{cheap}$	$20.5_{3.2}$	$31.1_{3.2}$	$35.6_{1.8}$	$30.5_{4.6}$

Table 16: RoBERTa$_{full}$ and RoBERTa$_{freeze}$ start out as the strongest classifiers on the first step, but are overtaken on subsequent steps by LaGoNN$_{exp}$, which ends up as strongest method overall.

Method	**Toxic Conversations**			
Moderate	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$34.2_{3.4}$	$45.5_{1.9}$	$52.4_{3.3}$	$45.7_{5.6}$
SetFit$_{exp}$	$33.6_{2.9}$	$47.2_{2.2}$	$46.6_{3.3}$	$44.3_{4.3}$
LaGoNN$_{exp}$	$\mathbf{36.6}_{4.2}$	$48.2_{2.7}$	$49.9_{3.7}$	$48.0_{4.4}$
SetFit$_{lite}$	$33.6_{2.9}$	$52.6_{2.0}$	$55.1_{1.6}$	$48.8_{7.3}$
LaGoNN$_{lite}$	$\mathbf{36.6}_{4.2}$	$\mathbf{56.1}_{1.5}$	$\mathbf{57.7}_{1.4}$	$\mathbf{52.3}_{6.8}$
RoBERTa$_{freeze}$	$34.2_{3.4}$	$38.4_{2.1}$	$39.5_{1.8}$	$38.0_{1.5}$
kNN	$19.4_{1.9}$	$21.5_{3.4}$	$22.4_{2.9}$	$21.6_{0.8}$
SetFit	$33.6_{2.9}$	$39.2_{2.9}$	$41.6_{2.7}$	$38.6_{2.4}$
LaGoNN	$\mathbf{36.6}_{4.2}$	$42.7_{3.7}$	$45.0_{3.5}$	$42.0_{2.5}$
Probe	$29.0_{2.7}$	$36.1_{1.2}$	$39.1_{1.5}$	$35.5_{3.3}$
LaGoNN$_{cheap}$	$26.1_{2.7}$	$34.3_{1.3}$	$37.5_{1.8}$	$33.6_{3.6}$

Table 17: On the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest, but it is LaGoNN$_{lite}$ that remains performant for all other steps. LaGoNN$_{lite}$ is also the strongest overall method based on the average.

Method	**Toxic Conversations**			
Balanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$32.3_{1.1}$	$42.7_{1.8}$	$54.1_{3.4}$	$43.8_{6.3}$
SetFit$_{exp}$	$35.7_{3.4}$	$32.6_{6.2}$	$37.4_{2.7}$	$36.5_{1.9}$
LaGoNN$_{exp}$	$\mathbf{40.4}_{4.4}$	$40.2_{6.6}$	$39.8_{7.5}$	$40.0_{1.2}$
SetFit$_{lite}$	$35.7_{3.4}$	$52.7_{2.5}$	$53.9_{2.2}$	$46.8_{7.8}$
LaGoNN$_{lite}$	$\mathbf{40.4}_{4.4}$	$\mathbf{52.9}_{2.6}$	$\mathbf{54.0}_{2.3}$	$\mathbf{48.3}_{6.4}$
RoBERTa$_{freeze}$	$32.3_{1.1}$	$39.2_{1.5}$	$41.0_{0.6}$	$38.5_{2.4}$
kNN	$17.4_{0.8}$	$23.7_{2.6}$	$24.3_{2.7}$	$23.1_{2.0}$
SetFit	$35.7_{3.4}$	$44.5_{2.9}$	$46.1_{2.8}$	$43.6_{2.9}$
LaGoNN	$\mathbf{40.4}_{4.4}$	$46.6_{2.7}$	$48.1_{2.2}$	$46.1_{2.2}$
Probe	$29.5_{2.4}$	$35.9_{0.9}$	$40.2_{0.9}$	$36.1_{3.5}$
LaGoNN$_{cheap}$	$26.8_{2.7}$	$34.5_{1.3}$	$38.5_{0.8}$	$34.4_{3.7}$

Table 18: On the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest, but it is LaGoNN$_{lite}$ that remains performant for all other steps. LaGoNN$_{lite}$ is also the strongest overall method based on the average.

Method		**Hate Speech Offensive**		
Extreme	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$30.2_{1.4}$	$43.5_{2.5}$	$\mathbf{51.2}_{2.2}$	$\mathbf{44.3}_{7.4}$
SetFit$_{exp}$	$30.3_{0.8}$	$\mathbf{44.0}_{1.3}$	$51.1_{2.0}$	$43.8_{6.5}$
LaGoNN$_{exp}$	$30.3_{0.7}$	$40.7_{2.9}$	$49.1_{4.4}$	$42.2_{6.2}$
SetFit$_{lite}$	$30.3_{0.8}$	$43.4_{2.5}$	$45.5_{3.4}$	$41.6_{4.6}$
LaGoNN$_{lite}$	$30.3_{0.7}$	$40.9_{3.4}$	$41.5_{4.8}$	$39.1_{3.6}$
RoBERTa$_{freeze}$	$30.2_{1.4}$	$33.5_{3.1}$	$34.4_{3.4}$	$33.1_{1.4}$
kNN	$\mathbf{31.5}_{1.2}$	$35.9_{2.7}$	$37.4_{2.0}$	$35.8_{1.7}$
SetFit	$30.3_{0.8}$	$38.4_{2.5}$	$41.1_{1.5}$	$37.8_{3.3}$
LaGoNN	$30.3_{0.7}$	$35.7_{2.6}$	$39.1_{2.4}$	$35.6_{2.7}$
Probe	$29.0_{0.2}$	$34.7_{1.5}$	$40.1_{2.1}$	$35.1_{3.8}$
LaGoNN$_{cheap}$	$29.0_{0.1}$	$36.9_{1.8}$	$40.5_{2.1}$	$36.2_{3.7}$

Table 19: kNN is the strongest method at first, but is overtaken by SetFit$_{exp}$ on the 5^{th} step, which is then overtaken by RoBERTa$_{full}$ on the 10^{th} step. RoBERTa$_{full}$ is overall most performant system based on the average.

Method		**Hate Speech Offensive**		
Imbalanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$50.6_{3.0}$	$65.2_{3.9}$	$\mathbf{70.3}_{1.2}$	$64.2_{5.3}$
SetFit$_{exp}$	$54.4_{4.3}$	$66.3_{1.8}$	$68.9_{2.0}$	$64.3_{4.5}$
LaGoNN$_{exp}$	$\mathbf{57.0}_{5.2}$	$\mathbf{67.0}_{4.4}$	$69.8_{2.1}$	$\mathbf{64.9}_{4.6}$
SetFit$_{lite}$	$54.4_{4.3}$	$65.5_{3.0}$	$65.9_{3.5}$	$63.5_{3.9}$
LaGoNN$_{lite}$	$\mathbf{57.0}_{5.2}$	$66.6_{2.6}$	$66.6_{1.9}$	$64.3_{4.1}$
RoBERTa$_{freeze}$	$50.6_{3.0}$	$54.1_{1.6}$	$55.3_{2.3}$	$54.1_{1.3}$
kNN	$55.6_{4.8}$	$57.3_{2.3}$	$58.8_{3.6}$	$57.4_{1.1}$
SetFit	$54.4_{4.3}$	$57.0_{3.9}$	$58.2_{3.8}$	$57.2_{1.1}$
LaGoNN	$\mathbf{57.0}_{5.2}$	$58.2_{4.1}$	$58.3_{3.4}$	$58.3_{0.6}$
Probe	$46.5_{2.2}$	$57.8_{1.7}$	$60.3_{1.2}$	$56.5_{4.5}$
LaGoNN$_{cheap}$	$47.1_{1.3}$	$56.5_{2.2}$	$59.5_{2.5}$	$55.6_{3.8}$

Table 20: On the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest, and LaGoNN$_{exp}$ continues to be performant, but is overtaken on the 10^{th} step by RoBERTa$_{full}$. LaGoNN$_{exp}$ is the strongest overall method based on the average.

Method		**Hate Speech Offensive**		
Moderate	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$61.9_{3.4}$	$70.8_{1.0}$	$\mathbf{72.5}_{1.4}$	$69.9_{3.2}$
SetFit$_{exp}$	$\mathbf{64.3}_{4.2}$	$70.6_{2.4}$	$72.4_{0.5}$	$69.8_{2.8}$
LaGoNN$_{exp}$	$63.8_{4.9}$	$\mathbf{71.0}_{2.1}$	$72.3_{1.0}$	$\mathbf{70.0}_{3.0}$
SetFit$_{lite}$	$64.3_{4.2}$	$70.3_{2.2}$	$71.2_{2.1}$	$69.3_{2.3}$
LaGoNN$_{lite}$	$63.8_{4.9}$	$70.7_{1.4}$	$71.4_{1.0}$	$69.4_{2.5}$
RoBERTa$_{freeze}$	$61.9_{3.4}$	$63.2_{4.1}$	$64.1_{4.5}$	$63.2_{0.6}$
kNN	$\mathbf{64.3}_{4.0}$	$63.3_{2.9}$	$63.9_{2.5}$	$63.7_{0.4}$
SetFit	$\mathbf{64.3}_{4.2}$	$67.3_{3.2}$	$67.6_{2.3}$	$66.9_{1.1}$
LaGoNN	$63.8_{4.9}$	$65.0_{5.3}$	$66.7_{5.9}$	$65.3_{0.9}$
Probe	$55.6_{1.7}$	$63.8_{0.8}$	$66.1_{0.3}$	$63.2_{3.0}$
LaGoNN$_{cheap}$	$56.0_{3.6}$	$62.2_{1.4}$	$66.0_{0.9}$	$62.3_{2.9}$

Table 21: Similar to the imbalanced setting, on the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest, and LaGoNN$_{exp}$ continues to be performant, but is overtaken on the 10^{th} step by RoBERTa$_{full}$. LaGoNN$_{exp}$ is the strongest overall method based on the average.

Method		**Hate Speech Offensive**		
Balanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	$59.7_{3.5}$	$66.9_{1.2}$	$\mathbf{69.2}_{1.8}$	$\mathbf{66.4}_{2.7}$
SetFit$_{exp}$	$60.7_{1.3}$	$66.3_{1.6}$	$67.5_{0.9}$	$65.9_{2.2}$
LaGoNN$_{exp}$	$\mathbf{61.5}_{1.7}$	$66.4_{1.4}$	$67.7_{0.9}$	$66.1_{1.8}$
SetFit$_{lite}$	$60.7_{1.3}$	$66.3_{2.0}$	$66.5_{0.9}$	$65.1_{1.7}$
LaGoNN$_{lite}$	$\mathbf{61.5}_{1.7}$	$\mathbf{67.1}_{1.1}$	$67.3_{0.8}$	$66.0_{1.7}$
RoBERTa$_{freeze}$	$59.7_{3.5}$	$60.4_{2.7}$	$63.1_{2.3}$	$61.0_{1.3}$
kNN	$60.7_{1.3}$	$59.6_{2.8}$	$59.5_{2.5}$	$59.5_{0.5}$
SetFit	$60.7_{1.3}$	$62.5_{0.7}$	$63.4_{1.0}$	$62.3_{1.0}$
LaGoNN	$\mathbf{61.5}_{1.7}$	$62.8_{1.5}$	$64.2_{1.0}$	$63.0_{0.9}$
Probe	$54.9_{1.4}$	$58.5_{0.9}$	$60.9_{0.4}$	$58.7_{1.7}$
LaGoNN$_{cheap}$	$54.2_{2.3}$	$58.6_{0.6}$	$60.6_{0.5}$	$58.5_{1.8}$

Table 22: Similar to the moderate setting, on the first step, LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ start out the strongest, but RoBERTa$_{full}$ overtakes LaGoNN$_{lite}$ by the 10^{th} step. RoBERTa$_{full}$ slightly outperforms LaGoNN$_{lite}$ and LaGoNN$_{exp}$ as the overall strongest method based on the average.

Method		**LIAR**		
Extreme	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	**32.0**$_{2.7}$	**34.7**$_{2.9}$	35.1$_{4.3}$	33.7$_{1.0}$
SetFit$_{exp}$	31.2$_{3.8}$	30.4$_{3.1}$	31.8$_{2.9}$	31.5$_{0.7}$
LaGoNN$_{exp}$	30.6$_{4.7}$	30.3$_{2.0}$	31.3$_{2.0}$	31.1$_{0.6}$
SetFit$_{lite}$	31.2$_{3.8}$	32.7$_{3.8}$	33.5$_{4.2}$	32.7$_{0.8}$
LaGoNN$_{lite}$	30.6$_{4.7}$	31.8$_{3.9}$	32.4$_{2.7}$	31.6$_{0.6}$
RoBERTa$_{freeze}$	**32.0**$_{2.7}$	32.8$_{4.5}$	34.2$_{5.0}$	33.2$_{0.7}$
kNN	27.0$_{0.5}$	27.3$_{0.8}$	27.9$_{0.8}$	27.4$_{0.3}$
SetFit	31.2$_{3.8}$	33.7$_{5.1}$	**35.7**$_{5.1}$	**34.3**$_{1.6}$
LaGoNN	30.6$_{4.7}$	32.0$_{4.6}$	33.7$_{5.4}$	32.6$_{0.9}$
Probe	30.7$_{2.0}$	30.6$_{3.9}$	31.7$_{2.9}$	31.1$_{0.4}$
LaGoNN$_{cheap}$	30.7$_{2.0}$	30.5$_{3.8}$	31.4$_{2.6}$	31.0$_{0.4}$

Table 23: RoBERTa$_{freeze}$ and RoBERTa$_{full}$ start out performant and RoBERTa$_{full}$ continues to be until the 10^{th} step where it is overtaken by SetFit, which ends up being the strongest overall method.

Method		**LIAR**		
Imbalanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	31.4$_{3.2}$	35.8$_{2.6}$	**40.0**$_{4.3}$	36.2$_{2.4}$
SetFit$_{exp}$	**32.3**$_{4.5}$	35.9$_{3.1}$	36.4$_{2.2}$	35.2$_{1.1}$
LaGoNN$_{exp}$	**32.3**$_{4.6}$	35.7$_{3.4}$	36.5$_{2.3}$	35.7$_{1.4}$
SetFit$_{lite}$	**32.3**$_{4.5}$	35.6$_{2.7}$	37.4$_{2.6}$	35.8$_{1.6}$
LaGoNN$_{lite}$	**32.3**$_{4.6}$	35.2$_{2.4}$	36.6$_{2.7}$	35.5$_{1.3}$
RoBERTa$_{freeze}$	31.4$_{3.2}$	34.1$_{2.6}$	35.6$_{3.2}$	34.0$_{1.4}$
kNN	27.0$_{0.2}$	28.5$_{1.0}$	29.0$_{1.0}$	28.7$_{0.7}$
SetFit	**32.3**$_{4.5}$	**36.5**$_{3.1}$	38.5$_{3.4}$	**36.3**$_{2.0}$
LaGoNN	**32.3**$_{4.6}$	34.9$_{2.2}$	36.9$_{2.5}$	35.3$_{1.4}$
Probe	30.7$_{3.0}$	32.8$_{1.8}$	35.0$_{1.6}$	33.5$_{1.5}$
LaGoNN$_{cheap}$	30.4$_{3.0}$	32.9$_{1.8}$	35.4$_{1.7}$	33.5$_{1.7}$

Table 24: LaGoNN, LaGoNN$_{lite}$, LaGoNN$_{exp}$, SetFit, SetFit$_{lite}$, and SetFit$_{exp}$ start out as the most performant, but SetFit is the strongest on the 5^{th} step and RoBERTa$_{full}$ on the 10^{th}. Overall, SetFit is strongest method based on the average over all steps.

Method		**LIAR**		
Moderate	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	33.9$_{3.1}$	38.4$_{2.7}$	**43.9**$_{2.2}$	**39.5**$_{3.0}$
SetFit$_{exp}$	33.0$_{2.6}$	37.2$_{1.8}$	38.7$_{1.5}$	37.4$_{1.6}$
LaGoNN$_{exp}$	**34.1**$_{3.4}$	**38.7**$_{2.3}$	39.0$_{1.8}$	37.8$_{1.5}$
SetFit$_{lite}$	33.0$_{2.6}$	38.5$_{1.3}$	40.4$_{2.0}$	38.2$_{2.1}$
LaGoNN$_{lite}$	**34.1**$_{3.4}$	38.4$_{2.0}$	39.6$_{1.5}$	37.9$_{1.6}$
RoBERTa$_{freeze}$	33.9$_{3.1}$	35.3$_{2.6}$	36.8$_{2.2}$	35.4$_{1.0}$
kNN	29.2$_{0.8}$	29.7$_{1.5}$	30.0$_{0.6}$	29.8$_{0.3}$
SetFit	33.0$_{2.6}$	37.2$_{3.9}$	39.4$_{3.5}$	37.0$_{1.8}$
LaGoNN	**34.1**$_{3.4}$	37.0$_{3.1}$	38.6$_{3.0}$	36.8$_{1.3}$
Probe	31.6$_{1.1}$	34.7$_{2.5}$	37.0$_{2.5}$	34.9$_{1.7}$
LaGoNN$_{cheap}$	31.4$_{0.9}$	35.3$_{2.3}$	37.6$_{2.0}$	35.3$_{1.9}$

Table 25: LaGoNN, LaGoNN$_{lite}$, and LaGoNN$_{exp}$ are the most performant classifiers on the first step, while LaGoNN$_{exp}$ remains strong until the 10^{th} step where it is overtaken by RoBERTa$_{full}$. RoBERTa$_{full}$ is the overally strongest method if we aggregate over all steps.

Method		**LIAR**		
Balanced	1^{st}	5^{th}	10^{th}	Average
RoBERTa$_{full}$	33.8$_{2.1}$	**39.4**$_{2.4}$	**43.5**$_{1.7}$	**40.2**$_{3.2}$
SetFit$_{exp}$	**34.4**$_{2.3}$	36.7$_{1.7}$	37.0$_{1.3}$	36.5$_{1.1}$
LaGoNN$_{exp}$	33.8$_{1.8}$	34.2$_{2.7}$	37.2$_{1.9}$	36.2$_{1.4}$
SetFit$_{lite}$	**34.4**$_{2.3}$	38.7$_{2.3}$	40.3$_{2.8}$	38.0$_{2.1}$
LaGoNN$_{lite}$	33.8$_{1.8}$	37.6$_{2.0}$	39.4$_{2.8}$	37.2$_{1.9}$
RoBERTa$_{freeze}$	33.8$_{2.1}$	36.6$_{1.6}$	38.6$_{1.5}$	36.7$_{1.5}$
kNN	30.1$_{0.4}$	31.3$_{2.1}$	30.6$_{1.1}$	30.9$_{0.4}$
SetFit	**34.4**$_{2.3}$	38.3$_{2.5}$	40.0$_{2.0}$	37.9$_{1.6}$
LaGoNN	33.8$_{1.8}$	38.3$_{1.3}$	40.6$_{0.6}$	38.1$_{2.0}$
Probe	32.1$_{1.9}$	35.2$_{1.4}$	37.2$_{2.5}$	35.2$_{1.7}$
LaGoNN$_{cheap}$	31.9$_{1.9}$	36.0$_{1.0}$	37.5$_{2.5}$	35.7$_{1.8}$

Table 26: SetFit, SetFit$_{lite}$, and SetFit$_{exp}$ start out the strongest on the first step, but are overtaken by RoBERTa$_{full}$ on the 5^{th} which remains the most performant on the 10^{th} step and if we consider the average over all steps.

A.5 Additional results: general text classification

In this Appendix section, we provide additional results from our general text classification experiments in the main text, Section 6. Here we show results comparing LAGONN$_{lite}$ against SetFit$_{lite}$ and LAGONN$_{exp}$ against SetFit$_{exp}$, but we include results for one to five neighbors with LAGONN LABDIST, Figures 7 and 8, respectively. The measure is average precision for IMDB, macro-F1 elsewhere.

In general, the number of neighbors we consider does not appear to have a large impact on LAGONN 's predictive power and our method continues to be a more stable classifier than SetFit and can generally be expected to improve SetFit's performance. We also see that continued fine-tuning with the embedding model is only helpful for cases when the dataset has a relatively large number of labels. One exception to this is the case of Student Question Categories, where there are four labels. While it is clear that SetFit$_{lite}$ is a stronger model than LAGONN lite, if we consider the more expensive alternatives, the story changes; if we continue to fine-tune, the prediction curves are essentially the same, and LAGONN$_{exp}$ seems to have a slight edge on SetFit$_{exp}$ as we add training data.

LIAR, both the collapsed version we considered in our content moderation experiments and the original version (Orig Liar) we examine in our general text classification experiments here, seems to be a very difficult dataset. Adding examples or increased fine-tuning does not appear to consistently increase model performance. We observed this across all experimental settings and balanced regimes and is a sensible finding, as it should be very difficult to determine the truth of a specific statement without additional context.

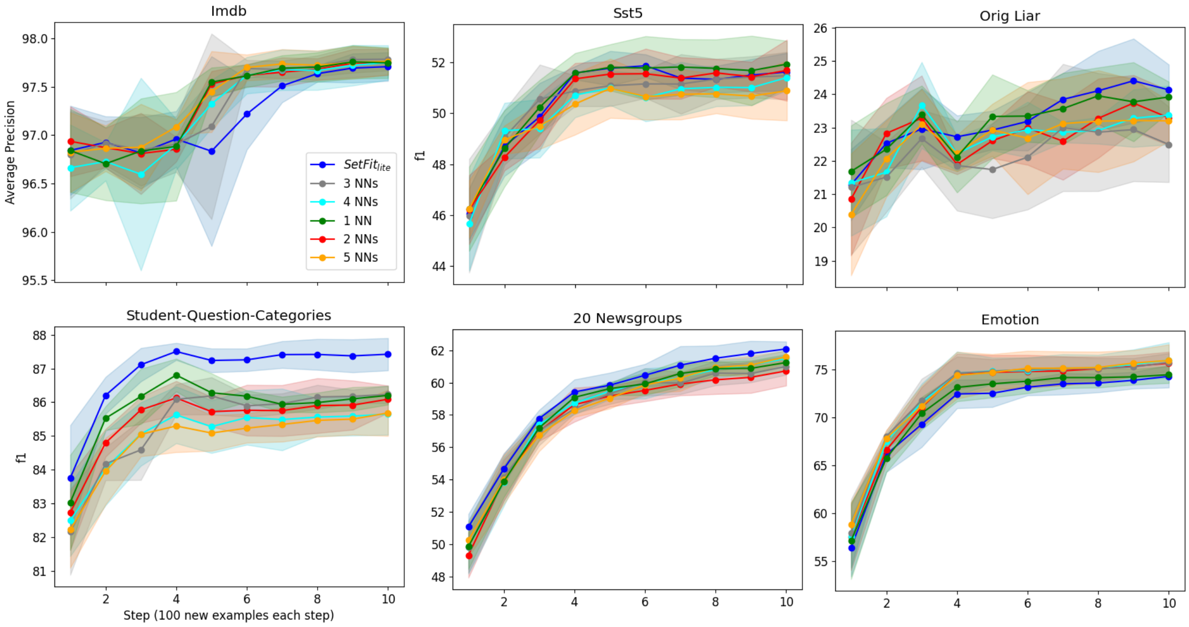

Figure 7: SetFit$_{lite}$ performance compared against one to five neighbors for LAGONN$_{lite}$ LABDIST. The measure is average precision for IMDB, macro-F1 elsewhere.

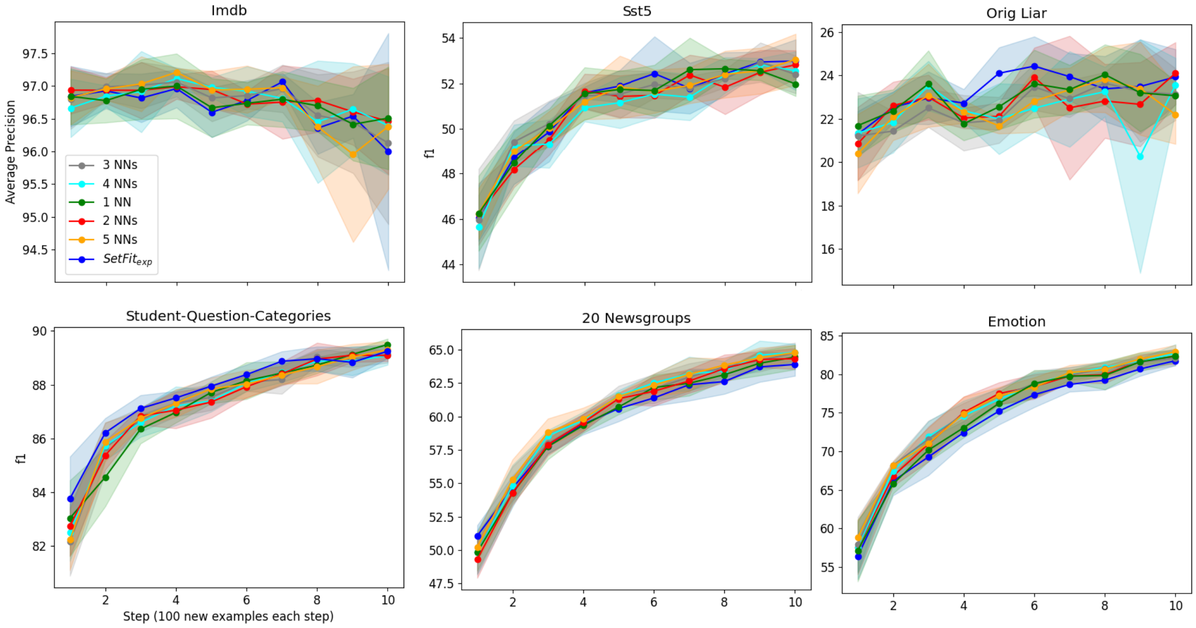

Figure 8: SetFit$_{exp}$ performance compared against one to five neighbors for LAGONN$_{exp}$ LABDIST. The measure is average precision for IMDB, macro-F1 elsewhere.

A.6 Additional results: multilingual text classification

In this Appendix section, we provide multilingual text classification results from experiments where we compare SetFit$_{exp}$ and SetFit$_{lite}$ against LaGoNN$_{exp}$ and LaGoNN$_{lite}$ respectively. For these experiments, we used the Multilingual Amazon Reviews Corpus (Keung et al., 2020), which has five labels, where each label is a star rating in Chinese, English, French, German, Japanese, or Spanish.[22] To create the mapping from label to text, we used code from the ADAPET (Tam et al., 2021) port in the official SetFit repository.[23] In these experiments, we used the same multilingual pretrained Sentence Transformer for all models under the balanced sampling regime.[24] In the case of LaGoNN$_{exp}$ and LaGoNN$_{lite}$, we use LabDist and search over one to five neighbors, reporting all results.

Figure 9 shows our results for expensive and inexpensive models. We note in all cases all models perform similarly. This supports our assertion in Section 6 that when the training data is balanced and we have only a handful of labels or less, it is sufficient to fine-tune the Sentence Transformer on only a subset of available training data. A classifier can then be fit on all available data, encoded with the fine-tuned ST. We observed this for SST-5 and observe it again here, especially clearly on the Chinese subset of this dataset. SetFit$_{exp}$ plateaus on the fifth step and stops learning, with different versions of LaGoNN$_{exp}$ outperforming it on later steps. However, if we move down on row, we see that all cheaper models continue to learn on all steps.

[22] https://huggingface.co/datasets/amazon_reviews_multi

[23] https://github.com/huggingface/setfit/blob/main/scripts/adapet/ADAPET/utilcode.py

[24] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

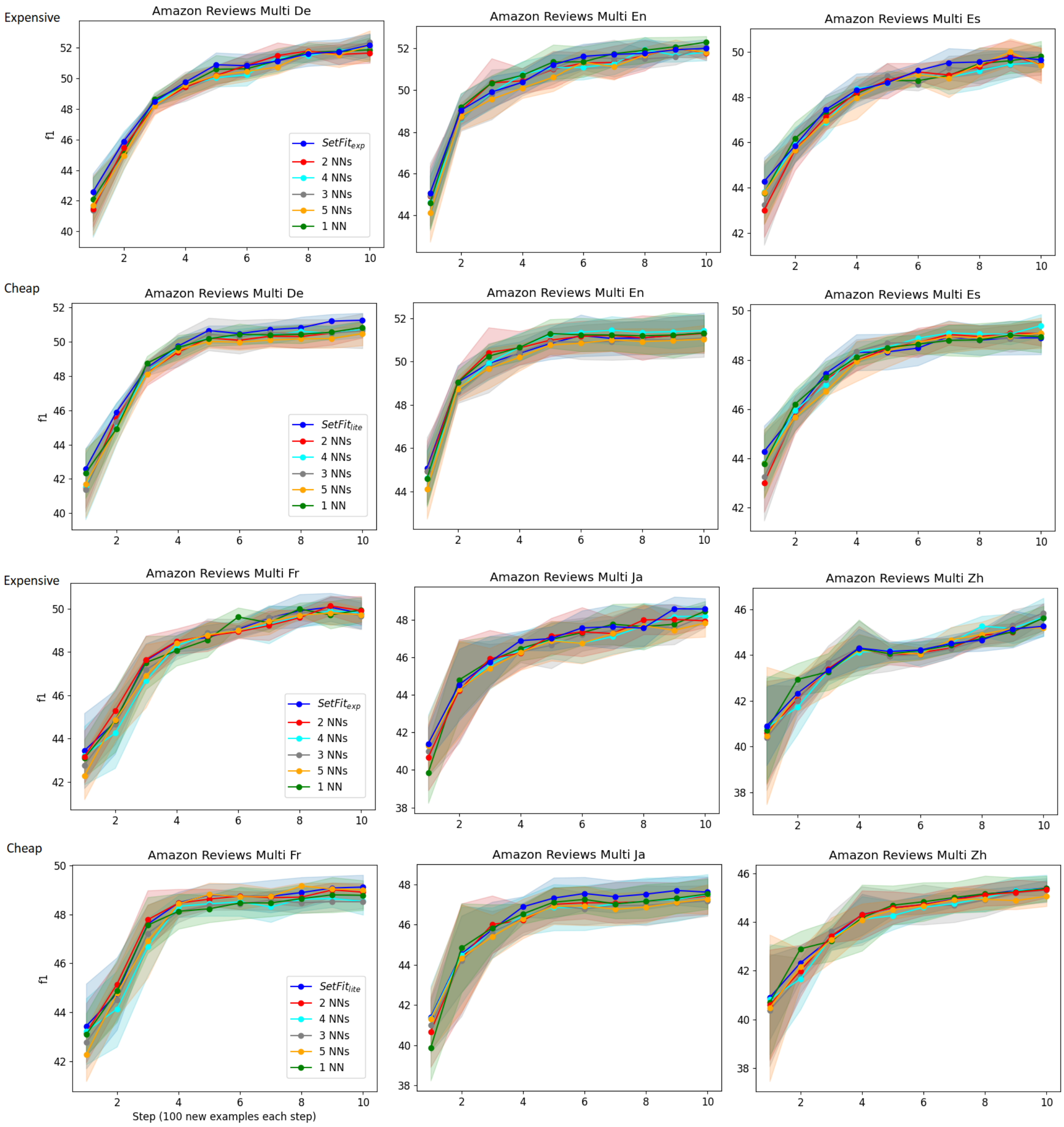

Figure 9: Multilingual classification experiments. In the first row, we display results from expensive models on German, English, Spanish data, with their cheaper counterparts in the following row. In the third and fourth row, we do the same but for French, Japanese, and Chinese. The measure is macro-F1 in all cases.