Adapting Language Models to Produce Reliable Class Probabilities in Classification Tasks

Anonymous authors
Paper under double-blind review

Abstract

Large generative language models (GLM) provide a versatile tool for solving a wide variety of natural processing tasks. GLM responses, though, are provided in the form of text, without an indication of the model's confidence in the answer. This limits the usability of these models on high-risk applications where decisions made based on an incorrect answer can have severe consequences. In this work, we focus on the problem of generating reliable class posterior distributions for text classification tasks, which can be used both for decision making and for producing interpretable confidence scores for the user. We show that the naive approach for computing posteriors based on the token posteriors produced by the GLM results in extremely poor posteriors. We then explore different adaptation approaches for improving the quality of posteriors, focusing on low resource scenarios where a small amount of data is available for adaptation. We show that parameter-efficient supervised fine-tunning (SFT), while providing large gains in terms of decision quality, produces suboptimal posteriors due to overfitting. To address this problem, we propose an approach that combines SFT and post-hoc calibration using a three-stage training strategy, improving the quality of both posteriors and categorical decisions.

1 Introduction

Modern neural language models provide an effective, adaptable and scalable tool for solving many natural language processing tasks. In particular, large generative language models (GLMs) like LLaMA 3 (Grattafiori et al., 2024), Qwen 2.5 (Yang et al., 2024) or Phi-4 (Abdin et al., 2024), are currently being used for a variety of complex natural language understanding tasks, showing outstanding performance on benchmarks related to reading comprehension, summarization, information retrieval, and generative question-answering (Narayan et al., 2018; Zellers et al., 2019; Khattab et al., 2023; Omar et al., 2023; OpenAI et al., 2024; Wei et al., 2021). While surprisingly good out-of-the-box performance can often be achieved with these models, they are generally unable to produce reliable measures of uncertainty for their answers, as shown in a large number of works (Lin et al., 2024; Cole et al., 2023; Estienne et al., 2023; Kuhn et al., 2022; Zhao et al., 2021). This limits the usability of these models for high-risk tasks where only very high confidence answers should be accepted (Zhang et al., 2021; Yoon et al., 2020).

For classification tasks, which are the focus of this work, decisions are often made based on class posterior probabilities using Bayes decision theory (Ferrer, 2024). For example, for the common case where all errors are assumed to cost the same, Bayes decisions correspond to the class with the highest posterior. The quality of the decisions then directly depends on the quality of the posteriors. Further, the posterior for the selected class is a measure of the confidence with which the system made the decision. This confidence score is valuable information that can be provided to the end user along with the decision. The goal of this work is to develop an approach for adapting GLMs to produce good-quality posterior probabilities for decision making and for interpretation by the end user.

Various approaches have been proposed for adapting GLMs to a specific domain. Standard techniques such as adding examples in the prompt to illustrate how the task should be solved or asking for a detailed explanation before producing an output are part of the family of methods called *prompt engineering* (or *prompting*, for short) (Liu et al., 2023). These methods have relatively low computational cost during the development phase because they do not require updating the parameters of the model. On the other hand, during deployment, they have the disadvantage of consuming tokens from the available prompt buffer and slowing down inference. Furthermore, the performance of the resulting system has been shown to strongly depend on the specifics of the prompt, making it a somewhat fragile approach (Zhao et al., 2021; Estienne et al., 2023). Another alternative for adapting a GLM to a downstream task or domain is supervised fine-tuning (SFT), which requires annotated data for the domain of interest. Due to the large size of modern GLMs, updating all of their parameters is computationally expensive and requires large amounts of adaptation data. To address this issue, a variety of parameter-efficient fine-tuning (PEFT) methods were proposed in the literature, among which LoRA (Edward J Hu et al., 2022) is currently the de-facto standard.

Although SFT methods are very effective for improving the performance of the model on the task of interest, our experiments show they do not ensure that the scores assigned to the predictions are good quality posterior probabilities for the task. Motivated by these results, we explore the use of post-hoc calibration (PHC) methods, which are approaches specifically designed to transform scores produced by a classification system into reliable posterior probabilities. In particular, we consider the family of affine logistic regression methods, which is widely used for this purpose and very cost-effective.

We evaluate the systems under comparison using two different metrics. To assess the quality of categorical decisions, we use the standard error rate, which assumes all errors to be equally costly. To assess the quality of posteriors we use proper scoring rules, as recommended by Ferrer & Ramos (2025). In contrast, recent machine learning papers concerned with this problem frequently use the expected calibration error (ECE) as a quality measure (Guo et al., 2017; Si et al., 2022; Ao et al., 2023). Yet, as discussed by Ferrer & Ramos (2025), the ECE and other calibration metrics do not adequately address the problem of assessing the value of posteriors since calibration only reflects one aspect of their performance (Gneiting & Raftery, 2007; Kull & Flach, 2015; Ferrer & Ramos, 2025). Hence, we instead use the cross-entropy, which, as the expectation of a proper scoring rule, is specifically designed for the problem of measuring the quality of posteriors (Winkler & Murphy, 1968; Gneiting & Raftery, 2007; Ferrer & Ramos, 2025).

1.1 Summary of our contributions

In this work we propose an effective way to combine SFT and PHC for text classification systems based on GLMs. The proposed approach, described in Section 5.3, provides the best quality posteriors (as measured by cross-entropy) and, as a consequence, also the best quality decisions (as measured by error rate) compared to using only SFT or PHC. Importantly, our results show that a careful training strategy is required to combine the two methods in order to achieve simultaneously low cross-entropy and error rate. Naively training PHC on the same adaptation data used to train SFT gives the same performance as SFT alone, while splitting the adaptation data for SFT and PHC training is suboptimal. In addition, we show that temperature scaling–the most common calibration approach in current machine learning literature–is often ineffective for calibration of GLM scores. A simple generalization to an affine transformation provides significantly better results, often competitive with those obtained with SFT alone. This is shown in sections 6.2 and 6.3. Finally, in Section 6.4, we show that, after adaptation with the combined approach, the results from a small LLaMA model are similar to those from a significantly larger Qwen model. These results suggest that suboptimal models become competitive after adaptation with our proposed method, though this conclusion will have to be verified by experimenting with a larger number of GLMs. The code needed to replicate the results in this work, along with output scores for all methods under comparison, can be found at at `https://anonymous.4open.science/r/llmcal-4FBE/Readme.md`.

2 Previous Works

Many methods for aligning a GLM to a specific task of interest rely purely on Input-Output prompting techniques (Brown et al., 2020). These methods adapt a generic GLM to a given task by carefully designing the prompt instructions (Raffel et al., 2020), selecting examples that illustrate how to solve the task (Brown et al., 2020) or retrieving additional information to provide necessary context (Khattab et al., 2023). More elaborate methods, such as Chain-of-Thought (CoT) (Wei et al., 2022), Tree of Thoughts (Yao et al., 2023) or BetterTogether (Soylu et al., 2024), optimize the prompt by iteratively generating and refining context to improve task performance. Notably, prompting techniques are limited by the finite context length and increase runtime during deployment with respect to simple prompts. In this work, we focus on simple prompts for clarity and efficiency, while noting that our proposed method is compatible with—and can be layered on top of—more elaborate prompting strategies such as the ones mentioned above.

SFT methods can adapt a single model to one or multiple tasks using in-domain adaptation data (Wei et al., 2021). Given the large size of modern GLMs, fine-tuning all of their parameters requires a substantial amount of data and computational resources (Chung et al., 2024). To mitigate this problem, parameter-efficient fine-tuning (PEFT, Lialin et al., 2024) techniques have been proposed in the last few years, including additive (Lester et al., 2021; Houlsby et al., 2019), selective (Donahue et al., 2014; Gheini et al., 2021), and reparametrization (Aghajanyan et al., 2021; Liu et al., 2024) variants. In particular, LoRA (Edward J Hu et al., 2022), a reparametrization method, has shown exceptional performance across numerous tasks. LoRA reduces the number of trainable parameters by replacing the linear operations inside the network with a low-rank approximation based on the hypothesis that the difference between the new and the old parameters lies in a low-rank space. Importantly, the adaptation cost of SFT methods is restricted to the fine-tuning stage. The resulting model can run faster than those that use prompt-engineering because the prompts do not need to describe the task. On the other hand, even when using PEFT methods, adapting a GLM requires more data than what is needed for prompting (Mosbach et al., 2023; Zhang et al., 2023; Edward J Hu et al., 2022). We will use a LoRA-based SFT method which provides strong empirical performance and a favorable balance between training cost and adaptation effectiveness.

For the specific case of text classification tasks, another family of adaptation approaches consists of introducing a PHC transformation to convert the posteriors generated by the model into new posteriors that are better matched to the task of interest. For example, Zhao et al. (2021) proposed transforming the posterior for each class by dividing it by the prior for that class, estimated by running the GLM with a content-free query. Similarly, Fei et al. (2023) propose dividing the posterior for each class by the average posterior for a query consisting of in-domain random words. These works implicitly assume that the prior class distribution for the task of interest is uniform. Since this assumption holds approximately or exactly for many text classification datasets, the methods work quite well for those cases. Estienne et al. (2023) formalized the problem of bias mitigation and showed that superior results can be obtained compared to the approach proposed by Zhao et al. (2021) by shifting the log posteriors with a bias term trained by minimizing the cross-entropy on adaptation data. Further, they showed that the bias term can also be successfully estimated without annotated data, relying only on knowledge of the class priors for the task of interest. In a recent work on PHC for NLP, Spiess et al. (2024) use a Platt scaling calibration trained on labeled data to determine whether a code generated by a GLM is correct or not. In this work, we explore various PHC approaches and combine them with SFT, leveraging their ability to produce better posteriors while incurring negligible additional computational cost during inference.

Very few of these works on text classification focus on the problem of producing good quality posteriors, concentrating, instead, on the quality of the categorical decisions measured through accuracy, error rate, F1-score or precision/recall (Yin et al., 2019; Schopf et al., 2023; Gera et al., 2022; Gretz et al., 2023). Works that are interested in the quality of the posteriors usually report performance in terms of calibration error (Desai & Durrett, 2020; Kapoor et al., 2024; Stengel-Eskin et al., 2024) which, as mentioned above, is a problematic practice since these metrics do not adequately reflect the quality of the posteriors. Very few works on text classification report performance in terms of PSRs and it is usually done in addition to ECE (Xie et al., 2024; Tian et al., 2023), often leading to ambiguous conclusions. For example, in Tables 2, 3, and 5 of the work done by Tian et al. (2023) we can see that the Brier scores (which is the expectation of a PSR) for the TriviaQA dataset are best for the baseline posteriors, but the ECE is best for the proposed calibration methods. This happens because the calibration methods degrade discrimination with the unintended effect that the overall quality of the posteriors is degraded. Faced with the need to select one of the two systems, one should choose the one with the lowest Brier score, not the one with the lowest ECE since those system's posteriors are poorer. Therefore, in this work we assess the quality of categorical decisions using error rate, and evaluate the quality of the posteriors using proper scoring rules (PSRs), which provide complementary assessment of the model.

Prompt $\rho(\mathbf{x}, y_2)$

```
<|im_start|>system
Determine the category of the article given by the user.<|im_end|>
<|im_start|>user
The Blair-Caldwell African American Research Library is a
branch of the Denver Public Library in Denver, Colorado,
in the United States that serves the Five Points
neighborhood.  [...] <|im_end|>
<|im_start|>assistant
Educational Institution
```

Prompt template

Sample $\mathbf{x}$

Class identifier

Figure 1: Example of a sequence $\rho(\mathbf{x}, y_2)$ for the Qwen2.5 model for a DBPedia sample, where $\mathbf{x}$ is the sample text taken from the dataset, and y_2 is the label corresponding to class 2 of the DBPedia dataset ("Educational Institution").

3 Computing text classification posteriors using GLMs

There are many ways to use GLMs for downstream tasks. The most common way is to simply prompt them with a question or instruction and have them generate an answer. In some scenarios, though, this approach may produce answers that do not satisfy the requirements of the task. One such case is text classification, where the goal is to annotate a certain text with a class label selected among a pre-defined set of possible classes. A GLM used as a free-text generator may not necessarily respond with one of the valid classes for the task. Further, with this approach, the GLM does not produce posterior probabilities for the classes which, as we have argued, may be a requirement for high-risk tasks. In this section, we describe a general and principled way to use GLMs for producing class posteriors for text classification tasks.

The first step is to determine an appropriate token sequence with which to query the GLM. In this work, we use the procedure formalized by Liu et al. (2023), where a model- and task-dependent template $\rho(\cdot)$ is used to create the sequence of tokens that make up the full sentence (a prompt followed by the answer). That is, given an input text $\mathbf{x}$ to be classified and a class $y_k \in \mathcal{Y} \triangleq \{y_1, \ldots, y_K\}$, we create a token sequence, $(p, a_k) \triangleq \rho(\mathbf{x}, y_k)$ which is a concatenation of the prompt p, consisting of the input text $\mathbf{x}$ and some additional instructions, and the answer $a_k = (a_k(1), \ldots, a_k(M_k))$, representing a name for the class y_k (see Figure 1 for an example). Details on the token sequence used for each dataset and each model can be found in Appendix A.

Given a sample text $\mathbf{x}$ we can then create a prompt $\rho(\mathbf{x}, y_k)$ for each possible class, y_k, and run the GLM to obtain the posteriors for each token $a_k(m)$ in the answer conditioned on the previous tokens, $P_{\text{LM}}(a_k(m) \mid a_k(m-1), \ldots, a_k(1), p)$. Then, we can compute the posterior for the answer given the prompt as

$$P_{\text{LM}}(a_k \mid p) = P_{\text{LM}}(a_k(1) \mid p) \prod_{m=2}^{M} P_{\text{LM}}(a_k(m) \mid a_k(m-1), \ldots, a_k(1), p). \tag{1}$$

Finally, we can define a *task* probability $P_{\text{Task}}(y_k \mid \mathbf{x})$ for producing the class label $y_k \in \mathcal{Y}$ given the input $\mathbf{x}$ as a normalized version of $P_{\text{LM}}(a_k \mid p)$ that sums to one over all the valid classes:

$$P_{\text{Task}}(y_k \mid \mathbf{x}) \triangleq \frac{P_{\text{LM}}(a_k \mid p)}{\sum_{j=1}^{K} P_{\text{LM}}(a_j \mid p)} \tag{2}$$

This is a natural way of using the score produced by the model to obtain a probability for each class. In particular, if each class is represented by a single token, Equation (2) is equivalent to applying the softmax operator to the vector obtained by stacking the logits of the tokens corresponding to each class. However, Equation (2) contemplates the general case where the classes may be represented by a sequence of more than one token.

This simple approach allows us to work around the problem posed by free-text generation, whereby two different sequences of tokens may correspond to the same class. For example, for the polarity classification task, when prompted with the instruction `'Determine if the following review is positive or` `negative'`, the GLM may respond `'The review is negative'` or just `'Negative'`. These two answers would result in different posteriors. We could attempt to force the GLM to produce only the class name in the answer by adding instructions in the prompt. Yet, this would result in long prompts, specially when there are a large number of classes or when each class has a long description, and it would not fully guarantee that the GLM will comply. Instead, we simply query the GLM with the a_k sequence, with the desired answer included and use the token posteriors of the answer to compute Equation (2).

4 Proper scoring rules for assessment of classification systems

In this work, we follow the literature on strict proper scoring rules (SPSRs), which were proposed decades ago for the assessment of the quality of posterior distributions (Winkler & Murphy, 1968; Gneiting & Raftery, 2007; Bröcker, 2009). SPSRs assess the quality of posteriors by measuring the quality of the Bayes decisions that can be made with them (Dawid & Musio, 2014; Brummer, 2010). An example of an SPSR is the Brier loss, for which the expectation over the data is called Brier score (BS) (Brier, 1950), a commonly used metric in some medical applications (Huang et al., 2020; Van Hoorde et al., 2015). Another SPSR is the negative log-loss (NLL). The cross-entropy (CE) is the expectation of the NLL, a metric widely used as an objective function for training deep neural network models, including GLMs (Wei et al., 2021; Raffel et al., 2020; Chung et al., 2024). As CE is an SPSR, minimizing it encourages the output of the models to be reliable posterior probabilities. In this work, we will use CE instead of BS because the former penalizes extremely wrong posteriors more heavily. While Brier loss has a maximum penalty of 1, the NLL can be infinite when a posterior of 0 is assigned to the correct class. We believe this is a desirable characteristic for high-stakes applications where some errors have extreme consequences.

While the CE is perhaps the most widely used loss for model training, it is much more rarely used as an evaluation metric. This may be partly due to the fact that its values are hard to interpret for being unbounded. Yet, the CE can be normalized to make it easily interpretable. Assuming an evaluation dataset with N samples, $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$, and posteriors $\mathbf{q}^{(i)} = (q_1^{(i)}, \ldots, q_K^{(i)})$ for each of those samples, with $q_k^{(i)} \triangleq P_{\text{Task}}(y_k \mid \mathbf{x}^{(i)})\ \forall k = 1, \ldots, K$, the normalized CE (NCE for short) is given by:

$$\text{NCE} = \frac{\sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}_{\{y^{(i)}=y_k\}} \log q_k^{(i)}}{\sum_{k=1}^{K} N_k \log(N_k/N)}, \tag{3}$$

where N_k is the number of times the class y_k appeared in the evaluation set (Ferrer, 2024). The normalization is done by dividing by the CE of the best naive system, one that always outputs the prior distribution of the classes, having no access to the input sample. This CE turns out to be the entropy of the prior distribution. Systems with NCE values close to 1.0 have a performance comparable to that of the best naive system.

In this work, as is usually done in machine learning literature, categorical decisions are made by selecting the class with the highest posterior probability. These decisions, commonly referred to as argmax decisions, can be shown to optimize the error rate (or, equivalently, the accuracy which is given by one minus the error rate). When decisions are made this way, the resulting error rate is also an expected PSR, although not a strict one. The error rate assumes that all incorrect decisions have the same cost. As with the CE, it can be normalized by the error rate of a system that makes always the same decision based solely on the class priors, without access to the input sample. The normalized error rate of argmax decisions is given by:

$$\text{NER} = \frac{\sum_{i=1}^{N} \mathbb{1}_{\{\text{argmax}_k q_k^{(i)} \neq y^{(i)}\}}}{\sum_{k=1}^{N} \mathbb{1}_{\{\text{argmax}_k N_k \neq y^{(i)}\}}}, \tag{4}$$

Importantly, although both metrics are PSRs, a low error rate does not necessarily imply a low CE. As we will show in the experimental section, while the system's outputs may be good for making argmax decisions-resulting in a low NER, they may still be poor posterior probabilities of the classes given the input, resulting in a high CE. Conversely, a lower CE does not directly imply a lower NER, though, in practice, this is very often the case. For these reasons, it is important to report both metrics when the quality of the posteriors is of interest, in addition to the quality of the decisions.

5 Adapting GLMs to produce reliable posteriors

In this section we describe the methods explored to adapt a GLM to a text classification task. We divide the methods into three categories: (1) supervised fine-tuning (SFT), (2) post-hoc calibration (PHC) and (3) combinations of the two approaches.

5.1 Supervised fine-tuning (SFT)

Standard LoRA-based SFT (Edward J Hu et al., 2022) is performed using a dataset of sequences created as described in Section 3 from the adaptation samples for the task of interest. We believe the main conclusions from this work should hold for alternative variants of LoRA and other SFT approaches. We leave this exploration for future work.

The loss for optimization is given by the token-level CE corresponding only to the tokens of the answer. That is, given an adaptation set $\{(p^{(i)}, a^{(i)})\}_{i=1}^{N}$ where each sample i consist of a sequence of tokens $p^{(i)}$ (the input prompt) and a sequence of tokens $a^{(i)}$ (the desired answer), the training loss is computed as

$$\mathcal{L}_{\mathrm{LM}}(\boldsymbol{\theta}) = -\frac{1}{\sum_{i=1}^{N} L_i} \sum_{i=1}^{N} \sum_{l=1}^{L_i} \log P_{\mathrm{LM}}(a^{(i)}(l) \mid a^{(i)}(l-1), \ldots, a^{(i)}(1), p^{(i)}; \boldsymbol{\theta}) \tag{5}$$

where L_i is the number of tokens in answer $a^{(i)}$, $\boldsymbol{\theta}$ is the set of trainable parameters, and $a^{(i)}(0)$ is an empty sequence. We also considered computing the loss over the tokens for the full sentence, the prompt and the answer. We found both versions to give similar results, with the one computed only over the answer being faster to compute. For that reason, all experiments in this work are done with the loss computed only over the answer.

When training or fine-tuning large models, it is often a good idea to do early stopping, monitoring the error on a validation set and stopping the training process when that loss stops improving. This prevents overfitting, which is particularly important when the amount of training data is small, as in our scenario of interest. To study the effect of holding-out data for early stopping, we evaluate three different approaches:

- **SFT-wo-val**: we use all the available samples for training until convergence based on the training loss.
- **SFT-w-val**: we leave a portion of the adaptation set for validation, and stop training based on the CE loss on this set.
- **SFT-retrain**: we perform SFT with validation, as above, and keep track of the optimal number of steps. Then, we rerun SFT on the full training set, stopping after that same number of steps.

Implementation details for each method can be found in Appendix B. After the GLM is fine-tuned, the class posteriors are computed using Equation (2), as for the unadapted system.

As we will see in the results, the best SFT approach in terms of NER is SFT-wo-val, while the best approach in terms of NCE is SFT-retrain. This poses an apparent dilemma. Which approach should we choose if we care both about having good argmax decisions and also good posteriors? Fortunately, as we will see, we do not need to settle for one or the other, since a combination of SFT and PHC leads to the best performance on both metrics.

5.2 Post-hoc calibration (PHC)

PHC approaches consist of adding a final stage to a classification system to transform its outputs, $\mathbf{q}$, into better class posteriors, $\tilde{\mathbf{q}}$ (Silva Filho et al., 2023; Ferrer & Ramos, 2025). In our case, the system's outputs are given by Equation (2), computed with token posteriors obtained from the unadapted or the fine-tuned GLM. One of the simplest calibration methods involves applying an affine transformation to the logarithm of the posteriors of the system, followed by a softmax transform to obtain a new set of posteriors:

$$\tilde{\mathbf{q}} = \mathrm{softmax}(\mathbf{A} \log(\mathbf{q}) + \boldsymbol{\beta}), \tag{6}$$

where $\mathbf{q}$, $\tilde{\mathbf{q}}$, and $\boldsymbol{\beta}$ are vectors of dimension K, the number of classes, and $\mathbf{A} \in \mathbb{R}^{K \times K}$. If the parameters $\mathbf{A}$ and $\boldsymbol{\beta}$ of the affine transformation are trained to minimize the cross-entropy, the resulting model will produce the best possible posteriors on the adaptation data. If the transformation does not overfit that data, then those posteriors will also be good on unseen data.

Instances of this approach are linear logistic regression, also known as Platt scaling for binary classification (Platt, 2000), an extension of Platt scaling for the multi-class case called direction-preserving (DP) calibration (Brummer & Van Leeuwen, 2006), and matrix, vector, and temperature scaling (Guo et al., 2017). In our experiments, we will use four version of this transformation:

- Vector Scaling (**PHC-VS**): Matrix $\mathbf{A}$ is assumed to be diagonal.
- Direction Preserving (**PHC-DP**): Matrix $\mathbf{A}$ is replaced by a scalar, α.
- Bias Only (**PHC-BO**): Same as PHC-DP, but with the weight α fixed at 1.
- Temperature Scaling (**PHC-TS**): Same as PHC-DP but with the bias term fixed at 0.

In all cases, the parameters are trained to minimize the cross-entropy of the training data. In each step, the value of the loss is computed using all training samples, and if the value did not decrease for 10 steps, the training is stopped. The general expression for the affine transform, with A being a full matrix leads to unstable results and is very prone to overfitting given a relatively small amount of adaptation samples. For this reason, results using a full A matrix are not included in this work.

As we will see in the results, PHC is less effective than SFT in terms of NER for all adaptation sizes. It is also inferior in terms of NEC for the larger adaptation sets. Yet, when the adaptation set is smaller, PHC approaches can, in some cases, result in better NCE than SFT. Fortunately, the combined method described below leads to the best performance on both metrics, consistently outperforming the best of both methods on almost all tested scenarios.

5.3 Combining PHC and SFT

SFT can lead to overfitting, even when using LoRA, causing the model to produce suboptimal (overconfident) class posteriors. This overfitting, though, can be solved by applying PHC on the posteriors produced by the model after SFT. However, combining these two approaches is not trivial. The naive approach would be to use the same adaptation samples to fine-tune the model and train the calibrator. This approach does not work since models trained with CE are already be well-calibrated on the samples used for training. Hence, training a calibration model using the posteriors produced by the fine-tuned model for the samples used for fine-tuning would lead to a calibration transform close to the identity function. What we wish to fix is the miscalibration that occurs on samples unseen during fine-tuning–i.e., samples with a score distribution that resembles the one we will see during deployment. To this end, ideally, we need two different adaptation sets, one for SFT, and one for PHC. Yet, given a small amount of adaptation data, splitting the data to train the two models is quite suboptimal.

To make the best possible use of the limited amount of adaptation data, we propose the following procedure for combining SFT and PHC:

1. Perform SFT on a subset of the adaptation set, holding out some samples for the next step, but training until convergence of the training loss.
2. Train the parameters of the calibrator on the samples held out from fine-tuning, using the posteriors for the model obtained above . These posteriors are unbiased since those samples were unseen for that model.
3. Perform SFT using all the adaptation samples until convergence. This is the model previously called SFT-wo-val.
4. Apply calibration to the predictions of the model from step 3, using the parameters from step 2.

We will refer to this procedure as “SFT-wo-val + PHC”. Note that this method can be implemented with all the PHC variants described above.

6 Experiments

We conducted experiments using the instruct-tuned versions of LLaMA3.2 (Grattafiori et al., 2024), with 1 billion parameters[1], and Qwen2.5 (Yang et al., 2024), with 7 billion parameters[2], on five different classification tasks. We first describe the experimental setup and compare the results using LLaMA3.2 for the various adaptation approaches considered in this work. Then, we explore the impact of the size of the adaptation set. Finally, we compare the results for the two LLMs.

6.1 Experimental set-up

We focus on downstream tasks that can be framed as standard text classification problems, including results for the following five open-access datasets:

- **SST2** (Socher et al., 2013) the Stanford Sentiment Treebank, which includes movie reviews annotated as either positive or negative. We used the GLUE version of the dataset (Wang et al., 2018), which does not contain publicly available labels for the test split, so we used only the training and validation sets. We used 400 samples for evaluation selected from the combination of the original train+validation sets to ensure consistency of the label frequency between training and evaluation sets.
- **AGNews** (Zhang et al., 2015): a collection of news articles grouped into four categories. Samples were downloaded from the `fancyzhx/ag_news` huggingface repository, which contains the original version of the dataset. We used 400 samples for evaluation selected after pooling the original train and test sets.
- **DBPedia** (Lehmann et al., 2015): a dataset derived from Wikipedia, where each article is categorized into one of 14 topics. Samples were downloaded from the `fancyzhx/dbpedia_14` huggingface repository, which contains the standard 14-category version of the dataset. We used 700 samples for evaluation sampled from the original test set.
- **20NewsGroups** (Lang, 1995): posts from online newsgroups categorized into 20 different topics. Samples were downloaded from the `SetFit/20_newsgroups` huggingface repository, which contains a curated variant of the original dataset. We used 800 samples for evaluation selected after pooling the original train and test sets.
- **Banking77** (Casanueva et al., 2020): customer service queries related to online banking, classified into 77 intent categories. Samples were downloaded from the `PolyAI/banking77` huggingface repository, which contains the original version of the dataset. We used 1000 samples for evaluation selected after pooling the original train and test sets.

Appendix A includes details of the prompt template, the class prior distribution, and the number of evaluation samples for each dataset.

We consider the scenario where a small amount of in-domain samples is available for adapting the model to the downstream tasks and study the impact of the adaptation set size in the performance of the different approaches. To this end, we generate adaptation datasets of varying sizes, randomly selecting N samples from the subset not used for evaluation. The value of N was computed as $N = K \cdot 2^T$, i.e., proportional to number of classes, K, with the factor being a power of 2. We initially tried using the same set of factors T for all tasks. Yet, we found that the same T resulted in very different trends for tasks with two classes versus tasks with many classes. In particular, while $T = 1$ was too small for SST-2 and AGNews, with 4 and 8 samples being insufficient for adaptation, 154 samples were enough to obtain reasonable results for Banking77. We then concluded that in order to obtain similar trends across datasets we needed to make T depend on the number of classes. Parameterizing T as $T'/\log(K)$ and then rounding the result to the nearest power of 2, resulted in similar trends across datasets for the same value of $T' \in \{4, 5, 6, 7, 8\}$. After selecting N adaptation samples, for the SFT methods that require a validation set, we further split those N samples, keeping 70% for fine-tuning and the rest for validation.

[1] `https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct`
[2] `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`

6.2 Comparison of adaptation methods

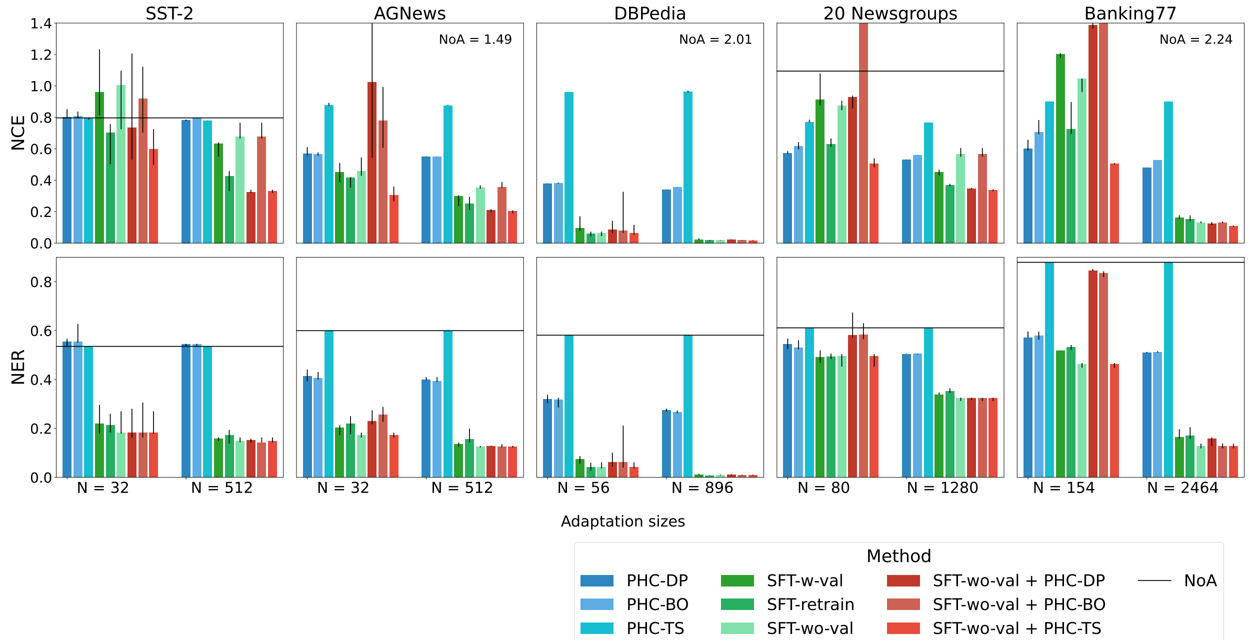

Figure 2: NCE and NER for all adaptation methods and two different adaptation samples (setting $T' = 4$ and $T' = 8$) on LLaMA3.2 model. Bars are divided by groups of colors: blue for PHC, green for SFT and red for SFT-wo-val + PHC methods. The bar height corresponds to the median across seeds used to select adaptation samples, and confidence intervals (black vertical lines on top of the bars) correspond to the 1st and 3rd quartiles. The NoA values are printed in the top right corner when they fall outside of the y-range.

Figure 2 shows the results for the LLaMA3.2 model after adapting it to each of the datasets above using the methods described in Section 5. Each plot shows two groups of bars, corresponding to the training sizes obtained with $T' = 4$ and $T' = 8$. Each bar shows the median NCE or NER across S adapted models obtained by sampling the adaptation data with S different random seeds. We used $S = 5$ seeds for DBPedia, 20Newsgroups and Banking77, for which the results were stable across seeds, and $S = 9$ seeds for SST-2 and AGNews, for which the results were noisier. Blue bars correspond to PHC, green bars to SFT and red bars to SFT-wo-val + PHC. Table 3 in Appendix C shows the numeric values corresponding to this figure.

Figure 2 shows that the decisions obtained with the unadapted model (NoA) perform relatively well, as indicated by NER values that are better than chance. On the other hand, the posteriors produced by the unadapted model are extremely poor, as indicated by NCE values that are worse than chance for all datasets except SST-2. Recall that, as explained in Section 4, the chance value for both metrics is 1.0, since they are both normalized by the value they would take for a naive system that only knows the class priors.

Comparing PHC methods, we can see that TS, the most common PHC method in current machine learning literature (Guo et al., 2017), is significantly worse than BO and DP calibration, both of which include a bias term. Among those two methods, DP is generally slightly better than BO in terms of NCE. Roughly speaking, the bias term compensates for mismatches in the prior class distribution between the model and the task of interest, while the weight compensates overfitting or underfitting problems (Ferrer & Ramos, 2025). It appears that a large source of miscalibration of the posteriors from this LLM is prior mismatch, since adding the bias term results in the largest gains from PHC. VS results are not shown in the figure since they are generally worse than DP, with some exceptions for the larger adaptation size where VS is better than DP (see Appendix C). Overall, we can see that PHC-DP already provides a large gain in performance for most datasets, compared to the unadapted baseline, specially for the NCE metric. This is particularly interesting given the advantage that this method has compared to SFT in tems of computational requirements.

Comparing now SFT methods, we can see that SFT-retrain is consistently better than or comparable to the other two SFT methods for the NCE metric. On the other hand, the best of the three SFT methods for the NER metric is SFT-wo-val, though the margin in this case is smaller than for NCE. Using all the available adaptation data for fine-tuning until convergence results in the best decisions. Yet, this approach results in overconfident posteriors due to overfitting. SFT-w-val reduces overfitting by stopping training based on validation performance, but at the cost of using fewer training samples. SFT-retrain aims to solve this problem by rerunning fine-tuning on all the data after determining the optimal number of training steps using a validation set, providing a significant improvement in NCE while slightly degrading the NER with respect to SFT-wo-val.

Overall, we can see that SFT gives consistently better results than PHC in terms of NER and, for the larger adaptation sets, also in terms of NCE. For the smaller adaptation size, though, SFT methods give poorer NCE results compared to PHC for 20 Newsgroup and Banking77, the two sets with larger class imbalance (see Appendix A). Again, this is probably due to overfitting since SFT, even when using LoRA, adapts a much larger number of parameters compared to PHC.

Finally, the red bars show the performance of the combined method, applying PHC on the SFT-wo-val posteriors, following the procedure of Section 5.3. For these posteriors, the best PHC approach is TS, supporting the hypothesis that SFT results in overfitting, which is easily reversed by the scaling of the log-posteriors performed by TS. On the other hand, correction of the priors is no longer needed after SFT, so the bias term is not required. Overall, SFT-wo-val followed by PHC-TS consistently gives the best results for both NCE and NER, eliminating the need for a trade-off between the two metrics. As mentioned in Section 5.3, training PHC on the same samples used to train SFT does not lead to improvements since the calibration transform learned on those samples is an identity function. These results are not shown in the plot since they coincide exactly with the SFT-wo-val results.

6.3 Impact of the adaptation set's size

Figure 3 shows the performance of the unadapted model (NoA) and selected adaptation methods for different number of adaptation samples, N, obtained by varying $T' \in \{4, 5, 6, 7, 8\}$. The selected methods are the best of each group except for SFT, for which we included two methods since, as observed above, one is the best for NER (SFT-wo-val) and the other one the best for NCE (SFT-retrain). The results for the left-most and right-most points in each plot coincide with those for the corresponding methods in Figure 2.

Overall, we can see the expected trend of degradation as the number of adaptation samples decreases for all methods, with a sharper degradation for SFT methods than for PHC-DP. When enough samples are available for adaptation, SFT is significantly better than PHC. However, as the number of samples gets smaller, PHC-DP outperforms SFT in two of the five datasets (20Newsgroups and Banking77).

The figure shows that conclusions from the previous section are consistent across all intermediate adaptation sizes: While SFT-retrain is better than SFT-wo-val for NCE, the trend is reversed for NER. Hence, if we had to select one SFT approach, we would have to prioritize one metric over the other. Further, in some scenarios, PHC-TS gives better NEC that both SFT approaches. Fortunately, the combined approach solves these trade-offs, reaching the best results across the board.

6.4 Comparison of results for LLaMA and Qwen models

In order to explore whether trends and conclusions hold across different LLMs, we run the main experiments using Qwen2.5-7B-Instruct, a model that is structurally quite different from the LLaMA models, of a different size and fine-tuned on instructions. Figure 4 shows a comparison of results for LLaMA3.2 and Qwen2.5 as a function of the number of adaptation samples. The linestyle identifies the model, while the methods are represented with the same colors as in Figure 3. Here we only include PHC-DP, SFT-wo-val and SFT-wo-val+PHC-TS for clarity.

The figure shows that, for simple tasks like SST-2 and AGNews, unadapted Qwen2.5 outperforms unadapted LLaMA3.2, both in terms of quality of decisions and of quality of posteriors. However, for more complex tasks like DBPedia, 20Newsgroups, and Banking77, Qwen2.5 slightly underperforms compared to LLaMA3.2,

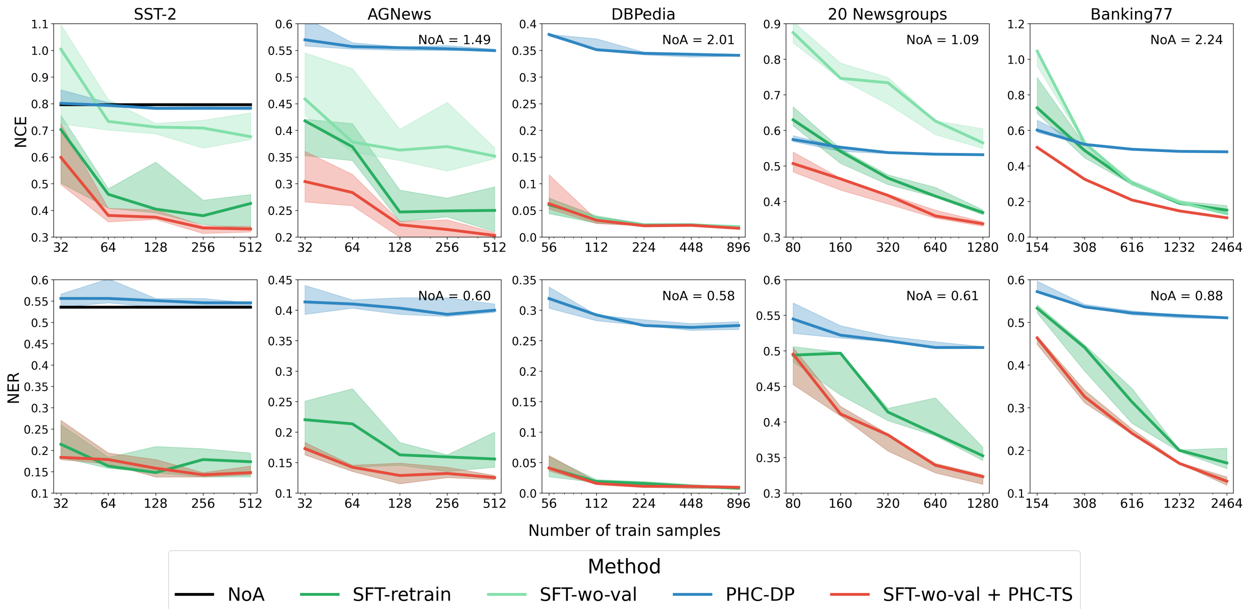

Figure 3: Performance of selected adaptation methods versus the number of adaptation samples on LLaMA3.2. Note that the NER curve for SFT-wo-val is behind the one for SFT-wo-val + PHC-TS, since TS does not change the NER value.

with both models performing considerably worse than the naive baseline in terms of NCE. Further, after SFT or the combined method, Qwen2.5 outperforms LLaMA3.2 consistently, though only by a relatively small margin. These results suggest that, after adaptation, different models converge to similar performance, despite being initially different.

Importantly, as shown in Figure 6 in Appendix C which presents the full results on the Qwen2.5 model, the main qualitative conclusions are the same as for the LLaMA model. PHC-DP outperforms PHC-BO and PHC-TS, in most cases giving large gains with respect to the unadapted model. The best SFT method for NCE is SFT-retrain, while SFT-wo-val has a small advantage for NER. Finally, the main result in this paper holds: SFT-wo-val followed by PHC-TS gives the best results of all adaptation methods across all combinations of dataset, adaptation size, and metric. These results support the hypothesis that the main conclusions in this work are robust to the choice of LLM.

7 Conclusions

In this work, we developed and compared various approaches for adapting GLMs to downstream classification tasks. Our main focus was to develop an approach that can produce reliable class posteriors without sacrificing the quality of the decisions. These posteriors could later be used to make optimal decisions using Bayes decision theory, or simply interpreted as a measure of the system's confidence. This is particularly important for high-stakes applications, where accepting an incorrect decision may have severe consequences.

We evaluated the performance of the resulting adapted models in terms of Normalized Error Rate (NER), which measures the quality of categorical decisions, and in terms of Normalized Cross Entropy (NCE), which measures the quality of the class posteriors. We compared simple post-hoc calibration approaches that transform the normalized sequence posterior from the GLM into better class posteriors and supervised fine-tuning using LoRA.

Our results show that the normalized sequence posterior from the unadapted model is an extremely poor class posterior, showing worse performance than a naive system that does not have access to the input text and only knows the class priors. Post-hoc calibration greatly improves the quality of the posteriors and also, as a

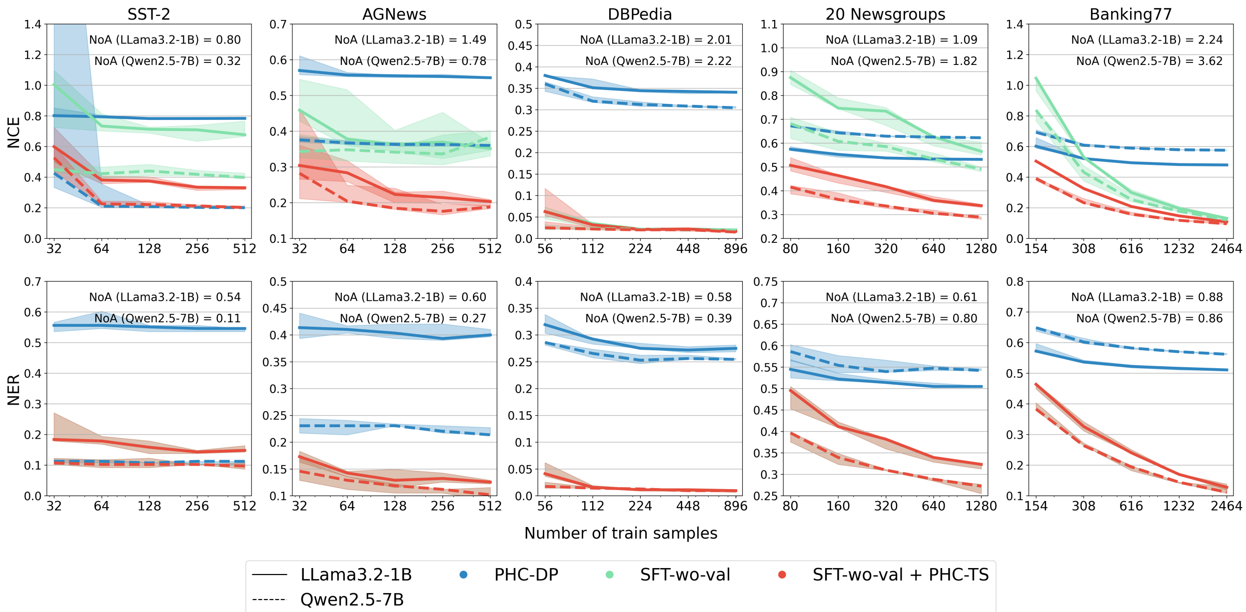

Figure 4: Results for LLaMA3.2 and Qwen2.5 models while varying the number of adaptation samples. Solid lines correspond to the LLaMA model, while dashed lines correspond to the Qwen model.

consequence, the quality of the categorical decisions, offering an efficient and effective adaptation approach. While fine-tuning leads to larger improvements than calibration in most cases, it is prone to overfitting, resulting in posteriors that overestimate the model's certainty. To address this issue, we proposed to apply calibration to the output of the fine-tuned model, developing an ad-hoc algorithm to optimally leverage the limited amount of adaptation data for training both stages. The model resulting from this combined adaptation approach consistently provides the best performance across all datasets for NCE and NER over a range of adaptation set sizes.

For future work, we plan to explore the integration of prompt engineering techniques for task adaptation, as we believe they could be complementary to the approaches studied in this work and further enhance model performance. In addition, we will expand this study to include more complex tasks like factual question answering, and information retrieval, where a posterior for correctness can be obtained with approaches similar to the ones studied in this work.

Broader Impact Statement and Limitations

The methods proposed in this work aim to improve the calibration of large language model (LLM) outputs, making their confidence scores more interpretable and reliable for downstream applications. This has the potential to enhance decision-making in critical areas such as healthcare, finance, and legal applications, where incorrect predictions could have grave consequences. By providing a more reliable measure of uncertainty, our approach could contribute to safer and more transparent AI systems.

It is important to note that the conclusions of this work are limited to cases where adaptation and evaluation scenarios are matched: the same prompt templates are used in both steps and the samples are extracted from the same dataset. Under these assumptions, we believe the proposed adaptation approach should work well across different models and datasets. If these assumptions do not hold – for example, if the prompt is changed to provide examples of the task during evaluation but not during adaptation – the quality of the posteriors for the adapted system is likely to degrade compared to the matched case. More research is required to determine the degree to which a model adapted using a given prompt or domain can be used for a different prompt or on a different domain.

References

Marah I. Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio CT Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 Technical Report, December 2024. URL `https://www.microsoft.com/en-us/research/publication/phi-4-technical-report/`.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning. 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)*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL `https://aclanthology.org/2021.acl-long.568/`.

Shuang Ao, Stefan Rueger, and Advaith Siddharthan. Two Sides of Miscalibration: Identifying Over and Under-Confidence Prediction for Network Calibration. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pp. 77–87. PMLR, July 2023. URL `https://proceedings.mlr.press/v216/ao23a.html`. ISSN: 2640-3498.

Glenn W. Brier. Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review*, January 1950. ISSN 1520-0493. URL `https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://papers.nips.cc/paper_files/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Niko Brummer. *Measuring, refining and calibrating speaker and language information extracted from speech*. PhD thesis, University of Stellenbosch, 2010. URL `https://scholar.sun.ac.za/items/1b46805b-2b1e-46aa-83ce-75ede92f0159`.

Niko Brummer and David A. Van Leeuwen. On calibration of language recognition scores. In *2006 IEEE Odyssey - The Speaker and Language Recognition Workshop*, pp. 1–8, June 2006. doi: 10.1109/ODYSSEY.2006.248106. URL `https://ieeexplore.ieee.org/abstract/document/4013523`.

Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society*, 135(643):1512–1519, July 2009. ISSN 0035-9009, 1477-870X. doi: 10.1002/qj.456. URL `https://rmets.onlinelibrary.wiley.com/doi/10.1002/qj.456`.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL `https://aclanthology.org/2020.nlp4convai-1.5/`.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and et al. Scaling Instruction-Finetuned Language Models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. ISSN 1533-7928. URL `http://jmlr.org/papers/v25/23-0870.html`.

Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively Answering Ambiguous Questions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 530–543, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.35. URL `https://aclanthology.org/2023.emnlp-main.35/`.

Alexander Philip Dawid and Monica Musio. Theory and applications of proper scoring rules. *METRON*, 72 (2):169–183, August 2014. ISSN 2281-695X. doi: 10.1007/s40300-014-0039-y. URL https://doi.org/10.1007/s40300-014-0039-y.

Shrey Desai and Greg Durrett. Calibration of Pre-trained Transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 295–302, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.21. URL https://aclanthology.org/2020.emnlp-main.21/.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 647–655. PMLR, January 2014. URL https://proceedings.mlr.press/v32/donahue14.html. ISSN: 1938-7228.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Lautaro Estienne, Luciana Ferrer, Matías Vera, and Pablo Piantanida. Unsupervised Calibration through Prior Adaptation for Text Classification using Large Language Models, August 2023. URL http://arxiv.org/abs/2307.06713. arXiv:2307.06713 [cs].

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. Mitigating Label Biases for In-context Learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14014–14031, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.783. URL https://aclanthology.org/2023.acl-long.783/.

Luciana Ferrer. No Need for Ad-hoc Substitutes: The Expected Cost is a Principled All-purpose Classification Metric. *Transactions on Machine Learning Research*, June 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=5PPbvCExZs.

Luciana Ferrer and Daniel Ramos. Evaluating Posterior Probabilities: Decision Theory, Proper Scoring Rules, and Calibration. *Transactions on Machine Learning Research*, August 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=qbrE0LR7fF.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. Zero-Shot Text Classification with Self-Training. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1107–1119, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.73. URL https://aclanthology.org/2022.emnlp-main.73/.

Mozhdeh Gheini, Xiang Ren, and Jonathan May. Cross-Attention is All You Need: Adapting Pretrained Transformers for Machine Translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1754–1765, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.132. URL https://aclanthology.org/2021.emnlp-main.132/.

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459, 1537-274X. doi: 10.1198/016214506000001437. URL http://www.tandfonline.com/doi/abs/10.1198/016214506000001437.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and et al. The Llama 3 Herd of Models, November 2024. URL http://arxiv.org/abs/2407.21783. arXiv:2407.21783 [cs].

Shai Gretz, Alon Halfon, Ilya Shnayderman, Orith Toledo-Ronen, Artem Spector, Lena Dankin, Yannis Katsis, Ofir Arviv, Yoav Katz, Noam Slonim, and Liat Ein-Dor. Zero-shot Topical Text Classification with LLMs - an Experimental Study. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9647–9676, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.647. URL https://aclanthology.org/2023.findings-emnlp.647/.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330. PMLR, July 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2790–2799. PMLR, May 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html. ISSN: 2640-3498.

Yingxiang Huang, Wentao Li, Fima Macheret, Rodney A. Gabriel, and Lucila Ohno-Machado. A tutorial on calibration measurements and calibration models for clinical prediction models. *Journal of the American Medical Informatics Association: JAMIA*, 27(4):621–633, April 2020. ISSN 1527-974X. doi: 10.1093/jamia/ocz228.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. Calibration-Tuning: Teaching Large Language Models to Know What They Don‘t Know. In Raúl Vázquez, Hande Celikkanat, Dennis Ulmer, Jörg Tiedemann, Swabha Swayamdipta, Wilker Aziz, Barbara Plank, Joris Baan, and Marie-Catherine de Marneffe (eds.), *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pp. 1–14, St Julians, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.uncertainlp-1.1/.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP, January 2023. URL http://arxiv.org/abs/2212.14024. arXiv:2212.14024 [cs].

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations*, September 2022. URL https://openreview.net/forum?id=VD-AYtP0dve.

Meelis Kull and Peter Flach. Novel Decompositions of Proper Scoring Rules for Classification: Score Adjustment as Precursor to Calibration. In *Machine Learning and Knowledge Discovery in Databases*, pp. 68–85, Cham, 2015. Springer International Publishing. ISBN 978-3-319-23528-8. doi: 10.1007/978-3-319-23528-8_5.

Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, ICML'95, pp. 331–339, San Francisco, CA, USA, July 1995. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-377-6.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, January 2015. ISSN 1570-0844. doi: 10.3233/SW-140134. URL https://content.iospress.com/articles/semantic-web/sw134. Publisher: IOS Press.

Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning, November 2024. URL http://arxiv.org/abs/2303.15647. arXiv:2303.15647 [cs].

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Contextualized Sequence Likelihood: Enhanced Confidence Scores for Natural Language Generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10351–10368, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.578. URL `https://aclanthology.org/2024.emnlp-main.578/`.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.*, 55(9):195:1–195:35, January 2023. ISSN 0360-0300. doi: 10.1145/3560815. URL `https://dl.acm.org/doi/10.1145/3560815`.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-Decomposed Low-Rank Adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 32100–32121. PMLR, July 2024. URL `https://proceedings.mlr.press/v235/liu24bn.html`. ISSN: 2640-3498.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot Fine-tuning vs. In-context Learning: A Fair Comparison and Evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12284–12314, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.779. URL `https://aclanthology.org/2023.findings-acl.779/`.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don‘t Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL `https://aclanthology.org/D18-1206/`.

Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. ChatGPT versus Traditional Question Answering for Knowledge Graphs: Current Status and Future Directions Towards Knowledge Graph Chatbots, February 2023. URL `http://arxiv.org/abs/2302.06466`. arXiv:2302.06466 [cs].

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and et al. GPT-4 Technical Report, March 2024. URL `http://arxiv.org/abs/2303.08774`. arXiv:2303.08774 [cs].

John C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10, June 2000.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. ISSN 1533-7928. URL `http://jmlr.org/papers/v21/20-074.html`.

Tim Schopf, Daniel Braun, and Florian Matthes. Evaluating Unsupervised Text Classification: Zero-shot and Similarity-based Approaches. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, NLPIR ’22, pp. 6–15, New York, NY, USA, June 2023. Association for Computing Machinery. ISBN 978-1-4503-9762-9. doi: 10.1145/3582768.3582795. URL `https://doi.org/10.1145/3582768.3582795`.

Chenglei Si, Chen Zhao, Sewon Min, and Jordan Boyd-Graber. Re-Examining Calibration: The Case of Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2814–2829, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.204. URL `https://aclanthology.org/2022.findings-emnlp.204/`.

Telmo Silva Filho, Hao Song, Miquel Perello-Nieto, Raul Santos-Rodriguez, Meelis Kull, and Peter Flach. Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Machine*

Learning, 112(9):3211–3260, Sep 2023. ISSN 1573-0565. doi: 10.1007/s10994-023-06336-7. URL https://link.springer.com/content/pdf/10.1007/s10994-023-06336-7.pdf.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Dilara Soylu, Christopher Potts, and Omar Khattab. Fine-Tuning and Prompt Optimization: Two Great Steps that Work Better Together. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10696–10710, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.597. URL https://aclanthology.org/2024.emnlp-main.597/.

Claudio Spiess, David Gros, Kunal Suresh Pai, Michael Pradel, Md Rafiqul Islam Rabin, Amin Alipour, Susmit Jha, Prem Devanbu, and Toufique Ahmed. Calibration and Correctness of Language Models for Code. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pp. 495–507. IEEE Computer Society, October 2024. ISBN 979-8-3315-0569-1. doi: 10.1109/ICSE55347.2025.00040. URL https://www.computer.org/csdl/proceedings-article/icse/2025/056900a495/215aWTgzWr6. ISSN: 1558-1225.

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. LACIE: Listener-Aware Finetuning for Calibration in Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024. URL https://openreview.net/forum?id=RnvgYd9RAh&referrer=%5Bthe%20profile%20of%20Mohit%20Bansal%5D(%2Fprofile%3Fid%3D~Mohit_Bansal2).

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

K. Van Hoorde, S. Van Huffel, D. Timmerman, T. Bourne, and B. Van Calster. A spline-based tool to assess and visualize the calibration of multiclass risk predictions. *Journal of Biomedical Informatics*, 54:283–293, April 2015. ISSN 1532-0464. doi: 10.1016/j.jbi.2014.12.016. URL https://www.sciencedirect.com/science/article/pii/S1532046415000027.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, October 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Robert L. Winkler and Allan H. Murphy. “Good” Probability Assessors. *Journal of Applied Meteorology and Climatology*, October 1968. ISSN 1520-0450. URL https://journals.ametsoc.org/view/journals/apme/7/5/1520-0450_1968_007_0751_pa_2_0_co_2.xml.

Johnathan Xie, Annie S Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. Calibrating Language Models with Adaptive Temperature Scaling. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18128–18138, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1007. URL https://aclanthology.org/2024.emnlp-main.1007/.

An Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, and et al. Qwen2.5 Technical Report, December 2024. URL http://arxiv.org/abs/2412.15115. arXiv:2412.15115 [cs].

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R. Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems*, November 2023. URL https://openreview.net/forum?id=5Xc1ecxO1h.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3914–3923, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1404. URL https://aclanthology.org/D19-1404/.

Wonjin Yoon, Jinhyuk Lee, Donghyeon Kim, Minbyul Jeong, and Jaewoo Kang. Pre-trained Language Model for Biomedical Question Answering. In *Machine Learning and Knowledge Discovery in Databases*, pp. 727–740, Cham, 2020. Springer International Publishing. ISBN 978-3-030-43887-6. doi: 10.1007/978-3-030-43887-6_64.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. SMedBERT: A Knowledge-Enhanced Pre-trained Language Model with Structured Semantics for Medical Text Mining. 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)*, pp. 5882–5893, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.457. URL https://aclanthology.org/2021.acl-long.457/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. Machine Translation with Large Language Models: Prompting, Few-shot Learning, and Fine-tuning with QLoRA. In *Proceedings of the Eighth Conference on Machine Translation*, pp. 468–481, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.wmt-1.43. URL https://aclanthology.org/2023.wmt-1.43/.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 12697–12706. PMLR, July 2021. URL https://proceedings.mlr.press/v139/zhao21c.html. ISSN: 2640-3498.

A Dataset details

A.1 Used prompts

Table 1 shows the structure of the prompt used as input to each model. Table 2 shows the instruction and list of class identifiers for each dataset.

Model	ρ(input text, y)
LLaMA3.2	<|begin_of_text|><|start_header_id|>system<|end_header_id|> instructions and task description<|eot_id|><|start_header_id|>user<|end_header_id|> input text<|start_header_id|>assistant<|end_header_id|> class identifier(y)
Qwen2.5	<|im_start|>system instructions and task description<|im_end|> <|im_start|>user input text<|im_end|> <|im_start|>assistant class identifier(y)

Table 1: Structure of the prompt for each model

A.2 Priors distribution

Figure 5 shows the prior distribution for each dataset. Classes were sorted by probability value. SST-2 and AGNews are almost perfectly balanced. For DBPedia and 20Newsgroup, the ratio between the prior for the most frequent and for less frequent class is around 2, while for Banking77 is around 3.

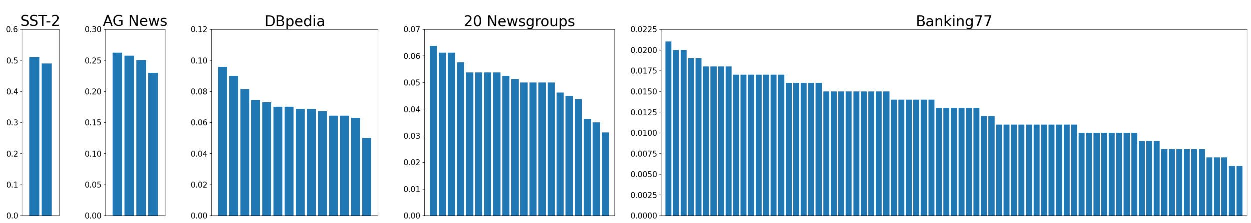

Figure 5: Prior distribution for each dataset. The class names are not included in the x-axis since the main goal is to show the class imbalance in each set.

B Implementation details

When using SFT in all its variants, we applied LoRA reparametrization to all attention, feed forward layers, and embeddings matrices of the model. LoRA hyperparameters were set to the default ones from the LitGPT library[3] ($p_{\text{dropout}} = 0.05$, $r = 8$, $\alpha = 16$). AdamW optimization was used, with a fixed learning rate of 10^{-3} and batch size of 8 for all datasets and adaptation sizes. Training and inference were done in bfloat16 precision.

For SFT-w-val, we select the model for which the validation cross-entropy is minimized using early stopping with a patience of 10. For SFT-retrain we proceed as follows: we keep track of the numbers of training steps used for fine-tuning for SFT-w-val. Then, we performed a second fine-tuning process with the same parameters and same number of steps, but using all available samples for training. Finally, for SFT-wo-val

[3]https://github.com/Lightning-AI/litgpt

Dataset	instruction template	a_k = class-identifier(y_k)
SST-2	Determine if the following review is positive or negative, based on the input given by the user.	*Positive; Negative*
AG News	Determine the category of the news article given by the user.	*World; Sports; Business; Science and Technology*
DBPedia	Determine the category of the article given by the user.	*Company; Educational Institution; Artist; Athlete; Office Holder; Mean Of Transportation; Building; Natural Place; Village; Animal; Plant; Album; Film; Written Work*
20Newsgroups	Determine the category of the posted document given by the user.	*Atheism; Graphics; Microsoft; IBM Hardware; Mac Hardware; X Window System; Sales; Cars; Motorcycles; Baseball; Hockey; Cryptography; Electronics; Medicine; Space; Christianity; Guns; Middle East; Politics; Religion*
Banking77	Classify the intent of the question input by the user.	*Active my card; Age limit; Apple pay or google pay; ATM support; Automatic top up; Balance not updated after bank transfer; Balance not updated after cheque or cash deposit; Beneficiary not allowed; Cancel transfer; Card about to expire; Card acceptance; Card arrival; Card delivery estimate; Card linking; Card not working; Card payment fee charged; Card payment not recognised; Card payment wrong exchange rate; Card swallowed; Cash withdrawal charge; Cash withdrawal not recognised; Change pin; Compromised card; Contactless not working; Country support; Declined card payment; Declined cash withdrawal; Declined transfer; Direct debit payment not recognised; Disposable card limits; Edit personal details; Exchange charge; Exchange rate; Exchange via app; Extra charge on statement; Failed transfer; Fiat currency support; Get disposable virtual card; Get physical card; Getting spare card; Getting virtual card; Lost or stolen card; Lost or stolen phone; Order physical card; Passcode forgotten; Pending card payment; Pending cash withdrawal; Pending top up; Pending transfer; Pin blocked; Receiving money; Refund not showing up; Request refund; Reverted card payment?; Supported cards and currencies; Terminate account; Top up by bank transfer charge; Top up by card charge; Top up by cash or cheque; Top up failed; Top up limits; Top up reverted; Topping up by card; Transaction charged twice; Transfer fee charged; Transfer into account; Transfer not received by recipient; Transfer timing; Unable to verify identity; Verify my identity; Verify source of funds; Verify top up; Virtual card not working; Visa or mastercard; Why verify identity; Wrong amount of cash received; Wrong exchange rate for cash withdrawal*

Table 2: Instructions and list of class identifiers for each dataset.

we trained the model using all the available samples until the training cross-entropy converged using the same criterion as for SFT-w-val.

We train all the PHC methods described in Section 5.2 with the LBFGS algorithm in a single batch per epoch until the training loss do not increase its value for 10 consecutive steps. Learning rate was set to 10^{-3} and the maximum number of iterations in each step of the LBFGS algorithm to 40.

C Additional results

C.1 Supplementary table for LLaMA3.2 model

Table 3 shows the numeric values corresponding to Figure 2. The numbers represent the NCE and NER for each method and dataset. These cases only correspond to the SFT-wo-val + PHC-VS of 20Newsgroups and Banking77.

		SST-2		AGNews		DBPedia		20 Newsgroups		Banking77	
		NCE	NER	NCE	NER	NCE	NER	NCE	NER	NCE	NER
	NoA	0.80	0.54	1.49	0.60	2.01	0.58	1.09	0.61	2.24	0.88
$T' = 4$	PHC-TS	**0.80**	**0.54**	0.88	0.60	0.96	0.58	0.77	0.61	0.90	0.88
	PHC-VS	0.86	0.57	0.58	0.42	1.08	0.43	0.73	0.55	1.34	0.60
	PHC-BO	0.81	0.56	**0.57**	**0.41**	**0.38**	**0.32**	0.62	**0.53**	0.71	0.58
	PHC-DP	**0.80**	0.56	**0.57**	**0.41**	**0.38**	**0.32**	**0.57**	0.54	**0.60**	**0.57**
	SFT-w-val	0.96	0.22	0.45	0.20	0.10	0.07	0.91	**0.49**	1.20	0.52
	SFT-retrain	**0.70**	0.21	**0.42**	0.22	**0.06**	**0.04**	**0.63**	**0.49**	**0.73**	0.53
	SFT-wo-val	1.00	**0.18**	0.46	**0.17**	**0.06**	**0.04**	0.88	0.50	1.05	**0.46**
	SFT-wo-val + PHC-TS	**0.60**	0.18	**0.30**	**0.17**	**0.06**	**0.04**	**0.51**	**0.50**	**0.50**	**0.46**
	SFT-wo-val + PHC-DP	0.73	0.18	1.03	0.23	0.08	0.06	0.93	0.58	1.39	0.84
	SFT-wo-val + PHC-BO	0.92	0.18	0.78	0.26	0.08	0.06	1.50	0.58	2.72	0.83
	SFT-wo-val + PHC-VS	1.18	**0.17**	2.03	0.28	0.13	0.06	-	0.59	-	0.78
$T' = 8$	PHC-TS	**0.78**	**0.54**	0.88	0.60	0.96	0.58	0.77	0.61	0.90	0.88
	PHC-VS	**0.78**	0.55	**0.49**	**0.39**	**0.25**	**0.24**	**0.51**	**0.49**	**0.45**	**0.50**
	PHC-BO	0.80	0.55	0.55	**0.39**	0.36	0.27	0.56	0.51	0.53	0.51
	PHC-DP	**0.78**	0.55	0.55	0.40	0.34	0.27	0.53	0.50	0.48	0.51
	SFT-w-val	0.63	0.16	0.30	0.14	**0.02**	**0.01**	0.45	0.34	0.16	0.17
	SFT-retrain	**0.43**	0.17	**0.25**	0.16	**0.02**	**0.01**	**0.37**	0.35	0.15	0.17
	SFT-wo-val	0.68	**0.15**	0.35	**0.13**	**0.02**	**0.01**	0.57	**0.32**	**0.13**	**0.13**
	SFT-wo-val + PHC-TS	0.33	0.15	**0.20**	**0.13**	**0.02**	**0.01**	**0.34**	**0.32**	**0.11**	**0.13**
	SFT-wo-val + PHC-DP	**0.32**	0.15	0.21	**0.13**	**0.02**	**0.01**	0.35	**0.32**	0.12	0.16
	SFT-wo-val + PHC-BO	0.68	**0.14**	0.36	**0.13**	**0.02**	**0.01**	0.57	**0.32**	0.13	**0.13**
	SFT-wo-val + PHC-VS	0.34	**0.14**	0.25	**0.13**	**0.02**	**0.01**	0.36	**0.32**	0.33	0.17

Table 3: Median value of NCE and NER for all adaptation methods corresponding to the LLaMA3.2 model for the smallest ($T' = 4$) and largest ($T' = 8$) adaptation size. The best performance for each task within each group is shown in bold, while the best overall performance across groups is shown in underlined bold. The values marked with "-" are cases in which PHC-VS diverged during training due to the small amount of training samples.

C.2 Results for Qwen2.5

Figure 6 shows the same results as Figure 2 but for the Qwen2.5 model. Equivalently, Figure 7 shows the same results as Figure 3 but for Qwen2.5. As explained in Section 6.4, this model shows better performance than LLaMA3 before adaptation and similar performance after adaptation with the proposed method. Finally, Table 3 is equivalent to Table 4 but for the Qwen2.5 model.

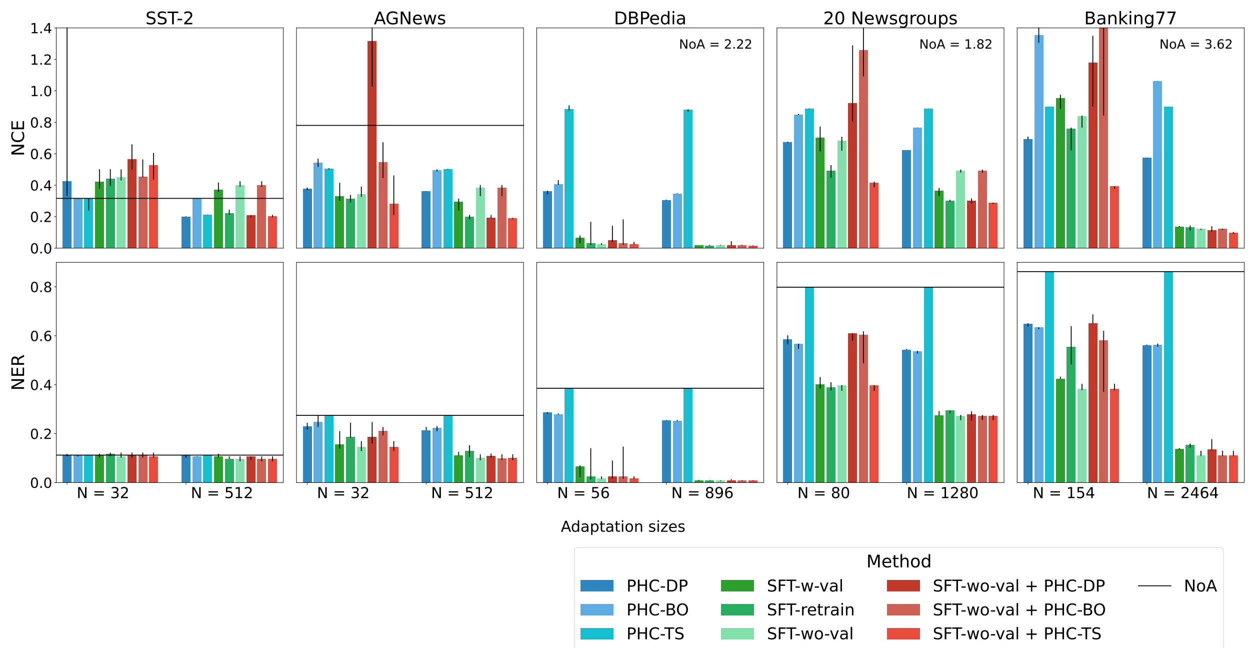

Figure 6: NCE and NER for all adaptation methods and two different adaptation samples ($T' = 4$ and $T' = 8$) on Qwen2.5 model. Bars are divided by groups of colors: blue for PHC, green for SFT and red for SFT-wo-val + PHC methods. The bar height corresponds to the median across seeds used to select adaptation samples, and confidence intervals (black vertical lines on top of the bars) correspond to the 1st and 3rd quartiles. Complete numerical results can be found in Table 4.

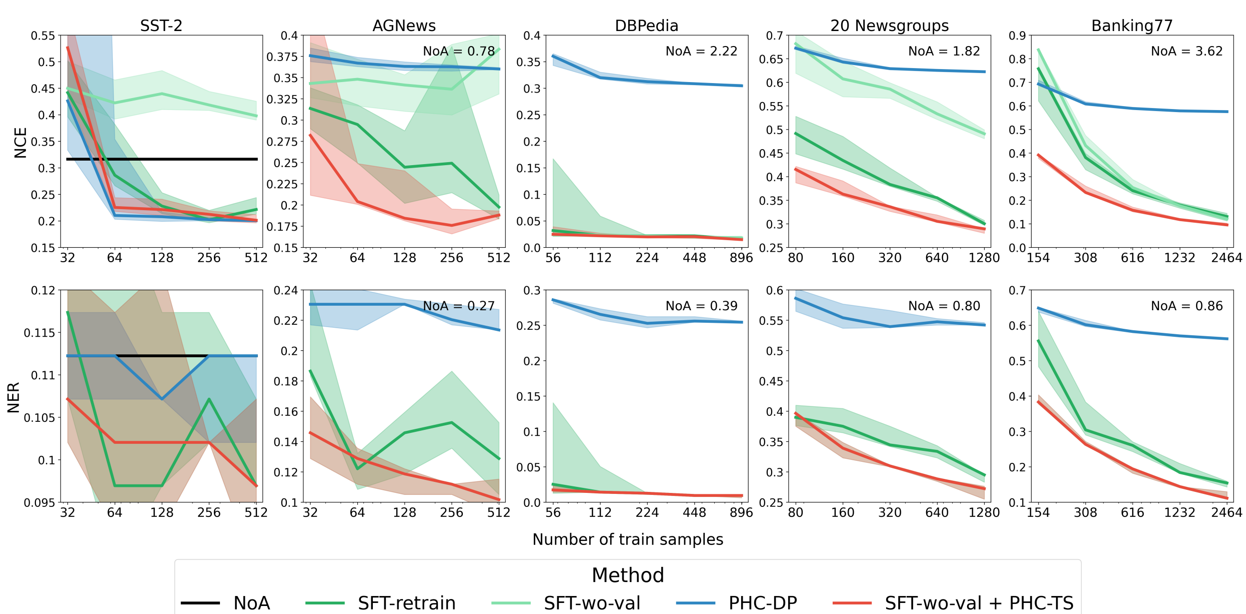

Figure 7: Performance of different adaptation methods versus the number of samples that were used for adapt on Qwen2.5. to the task. Methods shown in the plot are the most relevant from each group. The NER curves for SST-2 appear noisy but are almost constant (note the range of the y axis).

		SST-2		AGNews		DBPedia		20 Newsgroups		Banking77	
		NCE	NER	NCE	NER	NCE	NER	NCE	NER	NCE	NER
	NoA	0.32	0.11	0.78	0.27	2.22	0.39	1.82	0.80	3.62	0.86
$T' = 4$	PHC-TS	**<u>0.32</u>**	**<u>0.11</u>**	0.50	0.27	0.88	0.39	0.89	0.80	0.90	0.86
	PHC-VS	0.39	**<u>0.11</u>**	0.80	0.30	2.53	**0.23**	0.93	**0.57**	2.09	0.67
	PHC-BO	**<u>0.32</u>**	**<u>0.11</u>**	0.54	0.25	0.41	0.28	0.85	**0.57**	1.35	**0.63**
	PHC-DP	0.43	**<u>0.11</u>**	**0.38**	**0.23**	**0.36**	0.29	**0.67**	0.59	**0.69**	0.65
	SFT-w-val	**0.42**	**<u>0.11</u>**	0.33	0.16	0.06	0.07	0.70	0.40	0.95	0.42
	SFT-retrain	0.44	0.12	**0.31**	0.19	0.03	0.03	**0.49**	**<u>0.39</u>**	**0.76**	0.56
	SFT-wo-val	0.45	**<u>0.11</u>**	0.34	**<u>0.15</u>**	**<u>0.02</u>**	**<u>0.02</u>**	0.68	0.40	0.84	**<u>0.38</u>**
	SFT-wo-val + PHC-TS	0.53	**<u>0.11</u>**	**<u>0.28</u>**	**<u>0.15</u>**	**<u>0.02</u>**	**<u>0.02</u>**	**<u>0.42</u>**	**0.40**	**<u>0.39</u>**	**<u>0.38</u>**
	SFT-wo-val + PHC-DP	0.56	**<u>0.11</u>**	1.32	0.19	0.05	0.03	0.92	0.61	1.18	0.65
	SFT-wo-val + PHC-BO	**0.45**	**<u>0.11</u>**	0.55	0.21	0.03	0.03	1.26	0.60	1.43	0.58
	SFT-wo-val + PHC-VS	0.54	0.12	0.99	0.18	0.05	0.03	-	0.56	-	0.57
$T' = 8$	PHC-TS	0.21	**0.11**	0.50	0.27	0.88	0.39	0.89	0.80	0.90	0.86
	PHC-VS	**<u>0.20</u>**	**0.11**	**0.35**	**0.21**	**0.17**	**0.16**	**0.57**	**0.51**	**0.55**	0.57
	PHC-BO	0.32	**0.11**	0.50	0.22	0.34	0.25	0.77	0.54	1.06	**0.56**
	PHC-DP	**<u>0.20</u>**	**0.11**	0.36	**0.21**	0.30	0.25	0.62	0.54	0.58	**0.56**
	SFT-w-val	0.37	0.11	0.30	0.11	**0.02**	**<u>0.01</u>**	0.36	0.28	0.13	0.14
	SFT-retrain	**0.22**	**<u>0.10</u>**	**0.20**	0.13	**0.02**	**<u>0.01</u>**	**0.30**	0.30	0.13	0.15
	SFT-wo-val	0.40	**<u>0.10</u>**	0.38	**<u>0.10</u>**	**0.02**	**<u>0.01</u>**	0.49	**<u>0.27</u>**	**0.12**	**<u>0.11</u>**
	SFT-wo-val + PHC-TS	**<u>0.20</u>**	**<u>0.10</u>**	**<u>0.19</u>**	**<u>0.10</u>**	**<u>0.01</u>**	**<u>0.01</u>**	**<u>0.29</u>**	**<u>0.27</u>**	**<u>0.10</u>**	**<u>0.11</u>**
	SFT-wo-val + PHC-DP	0.21	0.11	**<u>0.19</u>**	0.11	0.02	**<u>0.01</u>**	0.30	0.28	0.11	0.13
	SFT-wo-val + PHC-BO	0.40	**<u>0.10</u>**	0.38	**<u>0.10</u>**	0.02	**<u>0.01</u>**	0.49	**<u>0.27</u>**	0.12	**<u>0.11</u>**
	SFT-wo-val + PHC-VS	0.21	**<u>0.10</u>**	0.24	0.11	0.02	**<u>0.01</u>**	0.34	0.28	0.42	0.14

Table 4: Median value of NCE and NER for all adaptation methods corresponding to the Qwen2.5 model for the smallest ($T' = 4$) and largest ($T' = 8$) adaptation size. The best performance for each task within each group is shown in bold, while the best overall performance across groups is shown in underlined bold. The values marked with “-” are cases in which PHC-VS diverged during training due to the small amount of training samples.