Enhancing In-context Learning via Linear Probe Calibration

Momin Abbas[⋆] **Yi Zhou**[†] **Parikshit Ram**[†]
Nathalie Baracaldo[†] **Horst Samulowitz**[†] **Theodoros Salonidis**[†] **Tianyi Chen**[⋆]
[⋆]Rensselaer Polytechnic Institute [†]IBM Research

Abstract

In-context learning (ICL) is a new paradigm for natural language processing that utilizes Generative Pre-trained Transformer (GPT)-like models. This approach uses prompts that include in-context demonstrations to generate the corresponding output for a new query input. However, applying ICL in real cases does not scale with the number of samples, and lacks robustness to different prompt templates and demonstration permutations. In this paper, we first show that GPT-like models using ICL result in unreliable predictions based on a new metric based on Shannon entropy. Then, to solve this problem, we propose a new technique called the *Linear Probe Calibration* (LinC), a method that calibrates the model's output probabilities, resulting in reliable predictions and improved performance, while requiring only minimal additional samples (as few as five labeled data samples). LinC significantly enhances the ICL test performance of GPT models on various benchmark datasets, with an average improvement of up to 21%, and up to a 50% improvement in some cases, and significantly boosts the performance of PEFT methods, especially in the low resource regime. Moreover, LinC achieves lower expected calibration error, and is highly robust to varying label proportions, prompt templates, and demonstration permutations. Our code is available at `https://github.com/mominabbass/LinC`.

1 Introduction

Large language models (LLMs), have remarkably showcased their capabilities across a broad range of natural language processing tasks (Devlin et al., 2018; Dong et al., 2019; Brown et al., 2020; Lewis et al., 2019; Yang et al., 2019; Radford et al., 2019). The cost of training these large models can be prohibitively expensive. Therefore, the commonly adopted approach is to first pre-train with large amounts of unlabled data and then fine-tune the model to downstream tasks. Although fine-tuning LLMs can be effective, it is prone to instability (Mosbach et al., 2021) due to numerous hyperparameter configurations resulting in failed runs, unstable results, and over-fitting (Raffel et al., 2020; Devlin et al., 2018; Kumar et al., 2022). Moreover, fine-tuning models of such large size may also be expensive and also requires explicit access to the architecture and weights of LLMs, which may not be publicly available (Zhang et al., 2022).

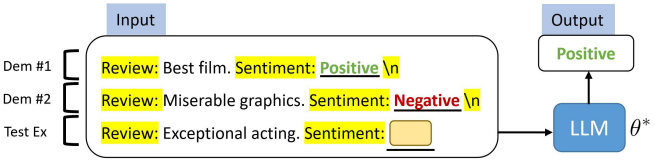

Figure 1: Example of ICL with a LLM θ^*.

To avoid large fine-tuning times and avoid requiring access to the weights of the model, recently, LLMs, exemplified by GPT-3 (Brown et al., 2020), have demonstrated the ability to perform *in-context learning* (ICL), a capability whereby a model can generate an appropriate output for a given query input based on a prompt that includes input-output example pairs specific to the task at hand. ICL can work with an API without explicit access to the LLM. Figure 1 provides a visual representation of ICL. The prompts utilized in ICL entail task-specific in conjunction with a series of input-label pairs, referred to as demonstrations. Such a capability of LLM to learn "in-context" presents an intriguing aspect whereby the model is capable of acquiring knowledge and performs well on a wide range of downstream tasks without any task-specific parameter fine-tuning (Brown et al., 2020; Touvron et al., 2023; Black et al., 2022; Rae et al., 2021).

More specifically, the aim of ICL is to make a prediction on some query test sam-

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

ple $\mathbf{x}$ by conditioning on a prompt sequence $(f_x(x_1), f_y(y_1), \ldots, f_x(x_k), f_y(y_k), f_x(\mathbf{x}))$ containing k-shot samples $(x_i, y_i)_{i=1}^k$ (i.e. demonstrations) and the query test sample, where the functions $f_x(.), f_y(.)$ denote template functions that attach pre-defined descriptions to the input and output, respectively (c.f. text highlighted in yellow in Figure 1). The output template function $f_y(.)$, in addition, may transform labels y_i into a natural language format instead into numeric/one-hot labels (e.g. for binary classification transforming labels $(0, 1)$ to (Positive, Negative) (c.f. labels in Figure 1)). Mathematically, for an input $\mathbf{x}$, a prompt P is defined as:

$$P(\mathbf{x}, (x_i, y_i)_{i=1}^k) \triangleq d_1 \oplus d_2 \oplus \cdots \oplus d_k \oplus f_x(\mathbf{x}) \quad (1)$$

where each demonstration d_i is given by $f_x(x_i) \oplus f_y(y_i)$ and $\oplus$ denotes the concatenation operation.

Recent studies suggest that ICL exhibits high variability in performance across different prompt templates, demonstrations, and their arrangement within the prompt arising from biases that favor outputting certain answers (Zhao et al., 2021; Lu et al., 2022), resulting in performance variation from random guess to state-of-the-art. Further, when the prompt P includes several more demonstrations, ICL's performance is thwarted by the inherent maximum sequence length limitation of the underlying language model. Moreover, our initial analysis reveals that while GPT-like models' ICL ability delivers acceptable results, their predictions cannot be considered reliable when assessed using Shannon entropy. We will discuss these limitations and challenges of ICL in detail in Section 2.

In this paper, we argue that by training very few parameters and subsequently utilizing an affine transformation on the output probabilities to *calibrate* the model, the performance, as well as the reliability of these predictions, can be significantly enhanced. We propose *linear probe calibration* (LinC), which optimizes the calibration parameters (i.e. a low-dimensional matrix and vector) using only a few extra samples and minimal computation. Experimental results reveal that LinC significantly outperforms the baselines. Moreover, LinC produces reliable predictions that exhibit consistency across a range of prompt templates and various permutations of demonstrations.

We summarize our contributions below:

C1) We present a novel insight that, while GPT-like models often exhibit acceptable ICL performance, their predictions appear to have very low confidence when measured using the Shannon entropy, highlighting a potential cause for their highly variable performance.

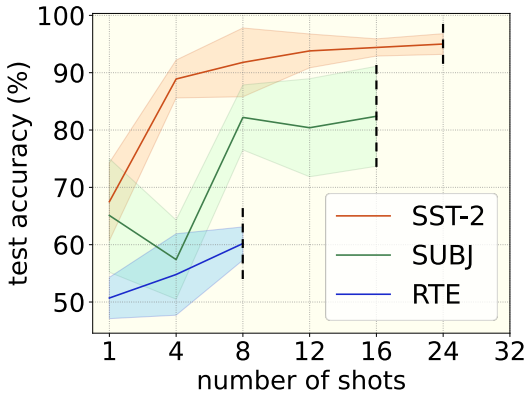

Figure 2: The efficacy of ICL is restricted by GPT tokenizer's maximum sequence length limit. Black-dashed lines demarcate the point beyond which additional shots cannot be utilized.

C2) We propose *linear probe calibration* (LinC), a simple and black-box method that enhances model's reliability and performance by linearly calibrating output probabilities without requiring any access to model weights or architecture.

C3) We empirically show that LinC consistently outperforms baselines on various benchmark datasets, with a performance boost of up to 21%, on average, and up to a 50% improvement in some cases, compared to the vanilla ICL baseline.

2 ICL: Challenges and Opportunities

In this section, we will first highlight two key limitations of ICL with current LLMs, which then serve as the motivation of our subsequent algorithm design.

2.1 Maximum Sequence length Limitation

We demonstrate the maximum sequence length limitation of ICL (Xu et al., 2023) on different datasets in Figure 2 using GPT-2 tokenizer. We observe that the test performance of ICL improves consistently, which aligns with the power-law relations between the generalization error and data size (Kaplan et al., 2020; Rosenfeld et al., 2020). However, beyond a certain point, no additional demonstrations can be added within the prompt since the model reaches its maximum sequence length limit[1]. Alas, the potential for performance enhancement through the acquisition of more examples is often constrained by this limitation of ICL, even in few-shot settings.

[1]It is noteworthy that although some recent models do have expanded context windows (e.g. GPT-4 with 32k tokens), the majority of them still maintain smaller context windows. Smaller models are often preferred in various applications due to their flexibility in adapting to specific tasks and their ability to achieve comparable performance to larger models while using much fewer compute.

Motivated by this, we ask:

Given a few additional samples, beyond the LLM sequence length limit, can we further improve ICL test performance without utilizing LLM fine-tuning?

2.2 Entropy of Few-Shot Learning with GPT

In practical decision-making systems, it is crucial for prediction networks to not only exhibit high accuracy but also have the ability to identify the likelihood of incorrect predictions. For example, automated healthcare systems should be designed such that when the confidence level of a disease diagnosis network is low, the control is transferred to human doctors for making the diagnoses (Jiang et al., 2012). Well-calibrated confidence estimates play a crucial role in making machine learning models more interpretable. Since humans possess an inherent cognitive understanding of probabilities (Cosmides and Tooby, 1996), having reliable confidence estimates can enhance the user's confidence in the model's predictions. This is especially relevant for neural networks, where the rationale behind their classification decisions can be complex and challenging to decipher. One way to assess the reliability of a model's predictions is to measure the *Shannon entropy*. Shannon entropy (Shannon, 1948) quantifies the amount of expected uncertainty in a probability distribution. Mathematically, Shannon entropy is given by:

$$E = -\sum_{c=1}^{C} \mathbf{p}_c \log(\mathbf{p}_c) \tag{2}$$

where C is the number of classes and $\mathbf{p}_c$ represents the output probability of class c. Entropy value is used to gauge prediction confidence, with a low value indicating high confidence and vice versa.

We compare the performance of ICL on GPT before (vanilla ICL) and after our calibration (Section 3) using different numbers of shots on various datasets. Figure 3 encapsulates the results on SST-2 (for other datasets, see Appendix B). We observe that, in contrast to the performance of our model, using vanilla ICL on GPT leads to high values of entropy, implying that most test predictions were made with very low confidence, i.e., close to random guessing. This observation indicates that although vanilla ICL (i.e., uncalibrated) on GPT yields satisfactory results in terms of test accuracy (refer to Table 4), the confidence associated with these predictions is not entirely reliable. These findings align with previous studies, including (Han et al., 2023), that have demonstrated the inadequacy of the conventional GPT decision boundary in effectively distinguishing between predictions by using the output with the highest probability as the predicted label. While low entropy (or high confidence) does not always imply a high accuracy, high entropy typically implies a high degree of uncertainty in predictions. This uncertainty may contribute to the increased variability of test performance, such as variations due to varying label proportions, prompt templates, and demonstration permutations (see Section 5.2). Motivated by this, we ask:

Can we make the predictions made by GPT-like models more reliable and accurate?

As we will see in the upcoming sections, our proposal to linearly calibrate model's output probabilities via meticulously learned parameters not only improves its test performance but also enhances its reliability.

3 Linear Probe Calibration

In order to be considered reliable, a model must furnish a *calibrated confidence* measure along with its predictions, where the probability assigned to the predicted class corresponds to its actual likelihood (Guo et al., 2017).

Since vanilla ICL predictions may be unreliable due to the preseznce of high entropy prediction probabilities, we aim to investigate whether probability calibration can lead to improvements. In this section, we present linear probe calibration (LinC) that calibrates the model's output probabilities, making it more reliable.

One method for modifying output probabilities applies an affine transformation (Platt, 2000; Guo et al., 2017):

$$\tilde{\mathbf{p}} = \mathsf{softmax}(\mathbf{A}\mathbf{p} + \mathbf{b}) \tag{3}$$

where $\mathbf{A}$ and $\mathbf{b}$ are parameters to be applied to the original probabilities $\mathbf{p}$ to get new probabilities $\tilde{\mathbf{p}}$. For example, for classification tasks $\mathbf{p}$ is the set of probabilities that are associated with each label[2]:

$$\mathbf{p}(P(\mathbf{x}, (x_i, y_i)_{i=1}^k)) \triangleq M_{\theta^*}(P(\mathbf{x}, (x_i, y_i)_{i=1}^k)) \tag{4}$$

for a model M_{θ^*} parameterized by θ^*.

Given a few additional samples, we then optimize $\mathbf{A}$ and $\mathbf{b}$ using a validation set, starting with a zero initialization; we find that our method exhibits remarkable insensitivity to initialization and works quite well for zero/random initialization (see Figure 10). More specifically, given a validation set $(x_i^{\mathrm{v}}, y_i^{\mathrm{v}})_{i=1}^{N_{\mathrm{v}}}$ of size N_{v}, we create N_{v} validation prompts via:

$$P_i^{\mathrm{v}} = P(x_i^{\mathrm{v}}, (x_j, y_j)_{j=1}^k). \tag{5}$$

[2]For detailed information about where the transformation is applied, please refer to Appendix B.

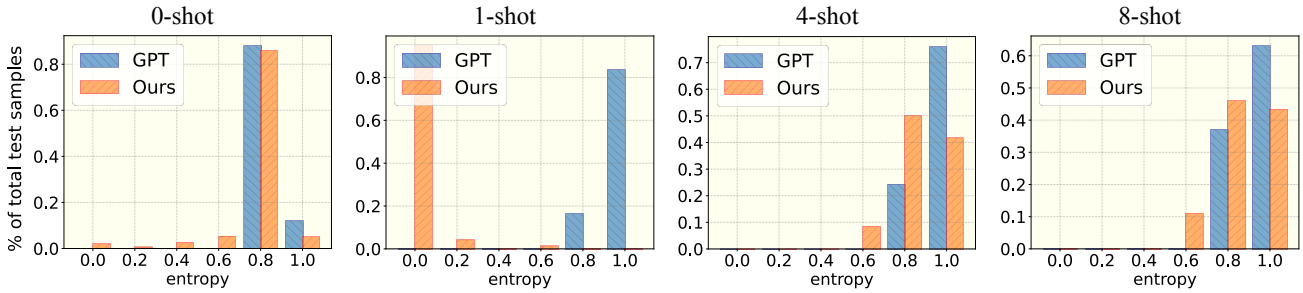

Figure 3: Shannon entropy histograms of using vanilla ICL on GPT-2-XL (1.5B) vs our method on SST-2 (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

Our objective is to solve the following problem

$$\min_{\mathbf{A},\mathbf{b}} \sum_{i=1}^{N_{\text{v}}} \mathcal{L}(\theta^*, \mathbf{A}, \mathbf{b}; P_i^{\text{v}}) \tag{6}$$

where $\mathcal{L}$ represents the loss function of the calibration parameters $\mathbf{A}$ and $\mathbf{b}$. We use a gradient-based optimizer to optimize $\mathbf{A}$ and $\mathbf{b}$ using prompts P_i^{v}. Algorithm 1 encapsulates our proposed LinC method. While we employ the stochastic gradient descent algorithm, our method can be utilized with any available optimization algorithm. LinC incurs negligible computational overhead and can be implemented in just a few lines of code to compute and store $\mathbf{A}$ and $\mathbf{b}$. Moreover, LinC utilizes only $k + N_v$ extra samples to learn $\mathbf{A}$ and $\mathbf{b}$. Finally, test predictions are obtained by calculating $\mathbf{Ap} + \mathbf{b}$ and taking the argument of the maxima after the softmax operator.

Comparison with other methods. Unlike calibration methods that rely on the raw data $(x_i^{\text{v}}, y_i^{\text{v}})_{i=1}^{N_{\text{v}}}$, our approach draws inspiration from ICL and employs prompts $\{P_i^{\text{v}}\}_{i=1}^{N_{\text{v}}}$ to learn the calibration parameters. Moreover, ICL performance is limited by the maximum input sequence length constraint of the underlying language model, causing poor scalability with an increase in the number of available training samples as shown in Figure 2. In contrast, our approach is scalable with respect to the number of available data samples, since we use the available samples to optimize the calibration parameters. In fact, our method requires only a handful of additional samples (typically in the range 10-100) to effectively optimize the calibration parameters, maintaining the few-shot regime and making it much more sample efficient than fine-tuning LLMs which require orders of magnitude larger number of samples. LinC is also much more computationally efficient as compared to other methods used to enhance LLM performance, such as fine-tuning, since it optimizes extremely low dimensional calibration parameters. Lastly, unlike in-context fine-tuning and meta-training methods (Min et al., 2022; Wei et al., 2022a), LinC operates using API access alone and doesn't need access to the LLM architecture and weights.

Algorithm 1 Linear Probe Calibration (LinC)

Input: LLM θ^*, validation set $(x_i^{\text{v}}, y_i^{\text{v}})_{i=1}^{N_{\text{v}}}$, k-shots $(x_j, y_j)_{j=1}^{k}$, loss function $\mathcal{L}$
Initialize: parameters $\mathbf{A}_0^0, \mathbf{b}_0^0$ via zero initialization, step-size α, number of epochs T
For each P_i^{v} in (5), obtain logits (4): $\mathbf{p}_i^{\text{v}} = \mathbf{p}(P_i^{\text{v}})$
for $t = 1, \cdots, T$ **do**
  for $i = 1, \cdots, N_{\text{v}}$ **do**
    Compute $\hat{\mathbf{p}}_i^{\text{v}} = \mathbf{A}_{t-1}^{i-1}\mathbf{p}_i^{\text{v}} + \mathbf{b}_{t-1}^{i-1}$
    $\mathbf{A}_{t-1}^{i} = \mathbf{A}_{t-1}^{i-1} - \alpha\nabla_{\mathbf{A}}\mathcal{L}(\hat{\mathbf{p}}_i^{\text{v}}, y_i^{\text{v}})$
    $\mathbf{b}_{t-1}^{i} = \mathbf{b}_{t-1}^{i-1} - \alpha\nabla_{\mathbf{b}}\mathcal{L}(\hat{\mathbf{p}}_i^{\text{v}}, y_i^{\text{v}})$
  end for
  $\mathbf{A}_t^0 = \mathbf{A}_{t-1}^{N_{\text{v}}}$ and $\mathbf{b}_t^0 = \mathbf{b}_{t-1}^{N_{\text{v}}}$
end for
return $\mathbf{A}_T^0, \mathbf{b}_T^0$

Shot	Method	SST-2	TREC	Subj
4-shot	NoC	$66.3_{13.1}$	$24.0_{6.3}$	$52.8_{4.9}$
	ConC	$78.9_{8.4}$	$41.3_{4.7}$	$67.8_{8.9}$
	NoC*	$68.9_{8.6}$	$35.9_{0.7}$	$69.3_{10.5}$
	ConC*	$75.9_{3.9}$	$43.5_{2.5}$	$61.9_{10.2}$
	LinC	$\mathbf{86.2}_{2.9}$	$\mathbf{45.9}_{3.3}$	$\mathbf{72.9}_{11.1}$
8-shot	NoC	$57.2_{8.0}$	$31.8_{8.1}$	$56.6_{10.7}$
	ConC	$73.7_{10.5}$	$45.4_{1.7}$	$68.1_{9.0}$
	NoC*	$62.7_{7.7}$	$40.5_{7.1}$	$70.4_{8.8}$
	ConC*	$75.8_{8.1}$	$47.2_{2.3}$	$63.0_{7.7}$
	LinC	$\mathbf{79.1}_{10.0}$	$\mathbf{48.0}_{5.4}$	$\mathbf{76.8}_{3.5}$

Table 1: Comparison under same number of samples.

4 LinC: A Viable Solution

Before presenting our results, we first highlight the significance of linear calibration and why it is important when we are handicapped by limited resources. Therefore we ask: *assuming that the maximum sequence length limit is not exceeded, what is the best way to use the additional samples in ICL? Is calibration the optimal way to use these additional samples?*

To answer this question, we introduce two additional baselines, NoC* and ConC*[3], which utilize the same 10 validation samples used for training the calibration parameters in our method LinC. For these baselines, these 10 samples are treated as additional in-context test demonstrations within the test prompts.

Consequently, NoC* and ConC* should be considered to be in the 14-shot and 18-shot regimes instead of the 4-shot and 8-shot regimes, respectively. The choice of using 10 samples is to ensure that the maximum length limit does not exceed for any of the datasets used in this experiment. Table 1 shows the results on three different datasets in the 4-shot and 8-shot settings on GPT-2-XL. It is noteworthy that for all three datasets, using the same samples to learn the calibration parameters is better than using them within the test prompts as additional test demonstrations.

Faster inference via calibration. LinC can be also viewed as speeding up the inference of in-context learning, particularly when the value of k is significantly large (e.g. when we would want to pack in as many demonstrations as we can in the prompt). If we have k-shot labeled demonstrations in the context, the inference time for vanilla ICL (i.e. NoC) would be $O(k^2)$ (removing the effect of the number of transformer blocks). Instead, if we use $n < k$ shot demonstrations in the context, the inference time for a new test point would be $O(n^2)$ with a speedup of $O((k/n)^2)$. Moreover, using the remaining $(k-n)$ demonstrations to learn the linear calibration parameters via SGD requires $O((k-n)n^2)$, which can be less than a single $O(k^2)$ inference with k demonstrations in the context. For example, if we choose $n = \sqrt{k}$, the calibration time is less than $O(k^2)$ while the inference speedup is $O(k)$.

Moreover, we encapsulate our finding in Figure 4 by assessing the trade-off between performance and resource consumption, quantified as the product of trainable parameters, samples, and epochs. The raw ICL baselines (0/8-shot) are denoted by horizontal lines. Notably, LinC outperforms ICL baselines by a substantial margin while maintaining resource efficiency. In addition, it enhances the performance of Parameter-Efficient Fine-tuning (PEFT) methods, such as Soft Prompt Tuning (SPT) (Lester et al., 2021) and Low-Rank Adaptation (LoRA) (Hu et al., 2022), particularly in scenarios with limited data and computational resources. These observations render LinC particularly effective in situations where data and compute resources are limited.

5 Experiments

This section will demonstrate the effectiveness of LinC on several benchmarks in the few-shot and PEFT settings. We run our experiments on GPT-2-XL (Radford et al., 2019) with 1.5B parameters, GPT-J (Wang and Komatsuzaki, 2021) with 6B parameters, and Llama-2 (Touvron et al., 2023) with 13B[4] parameters on 2 NVIDIA GeForce RTX 3090 GPUs. We thoroughly study different label proportions, prompt templates, and demonstration permutations.

[3]for details about NoC and ConC, see Section 5.1

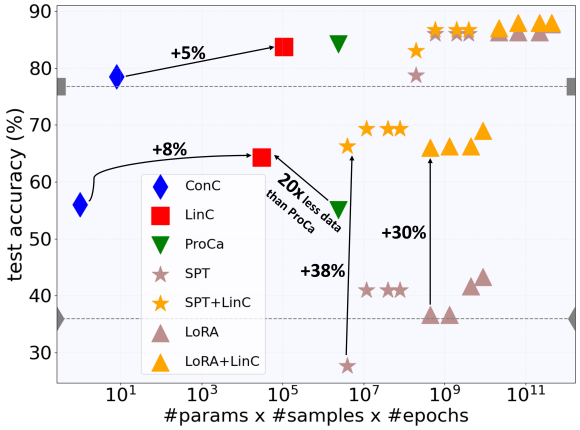

Figure 4: LinC outperforms ICL on all few-shot experiments, and substantially enhances PEFT, especially in the low resource regime, while maintaining almost identical data and compute requirements.

Dataset	GPT-J (6B)		
	ICL	**ConC**	**LinC**
SST-2	0.0592	0.1974	**0.0458**
SST-5	0.2254	**0.0933**	0.0970
AGNews	0.2858	0.0686	**0.0577**
TREC	0.2934	**0.1108**	0.1613
DBPedia	0.2713	0.2281	**0.0502**
RTE	0.0471	0.0825	**0.0421**

Table 2: Expected Calibration Error (ECE) comparison between baselines and LinC (a model with perfect calibration would exhibit an ECE of 0).

5.1 Experimental Setup

We evaluate the effectiveness of our LinC method using seven widely used text-classification datasets: sentiment analysis using SST-2 (Socher et al., 2013) and SST-5 (Socher et al., 2013), topic classification using the 4-way AGNews (Zhang et al., 2015) and 14-way DBPedia (Zhang et al., 2015), 6-way question classification using TREC (Voorhees and Tice, 2000), textual entailment using binary RTE (Dagan et al., 2005) from SuperGLUE (Wang et al., 2019), and subjectivity classification using Subj (Pang and Lee, 2004). We also tested LinC on seven diverse non-text classification

[4]Note that this is the largest model that can fit into our available GPU memory.

Dataset (D/C)	GPT-J (6B)		
	ICL	ConC	LinC
Hamster (5/2)	55.6±8.3	53.3±9.4	**60.0±14.4**
Customers (8/2)	67.4±0.5	54.9±2.3	**68.6±0.5**
Breast (7/2)	66.7±3.3	55.8±9.2	**71.3±3.3**
Spambase (57/2)	40.0±0.0	51.2±9.5	**61.1±1.7**
TAE (5/3)	45.2±4.6	48.4±9.5	**50.5±11.0**
Vehicle (18/4)	26.9±0.7	29.0±2.8	**29.0±1.5**
LED (7/10)	16.3±10.4	27.3±6.9	**28.0±7.1**

Table 3: Performance comparison between baselines and LinC on OpenML datasets; D represents the number of features and C represents the number of classes.

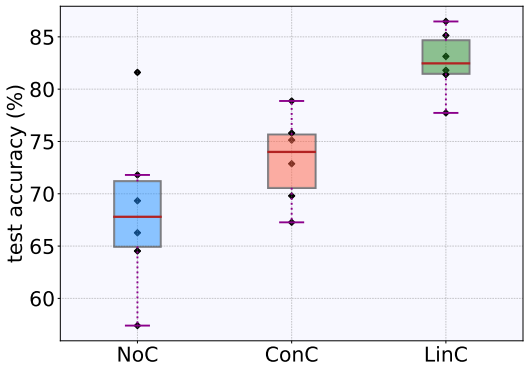

Figure 5: Comparison across six different templates.

tasks using OpenML (Vanschoren et al., 2014) datasets with varying number of classes and features.

A fixed prompt format was utilized for each dataset, unless stated otherwise, which is demonstrated alongside examples in Appendix A, Table 6 and Table 7.

Experiments were conducted under 0-shot, 1-shot, 4-shot and 8-shot learning settings. Five different sets of test demonstrations were chosen at random, and arranged in an arbitrary order in the prompt, and the mean and standard deviation were computed across all five test prompts. We used the cross-entropy loss as

$$\mathcal{L}(\theta^*, \mathbf{A}, \mathbf{b}; P_i^{\text{v}}, y_i^{\text{v}}) = -\sum_{c=1}^{C} y_{i,c}^{\text{v}} \log(\tilde{\mathbf{p}}_c) \qquad (7)$$

where $y_{i,c}^{\text{v}}$ is the binary variable indicating if class c is the correct label for validation input x_i^{v}, and $\tilde{\mathbf{p}}_c$ denotes the output from the softmax defined in (3) with $\mathbf{p} = \mathbf{p}(P_i^{\text{v}}) = \mathbf{p}(P(x_i^{\text{v}}, (x_j, y_j)_{j=1}^k))$ defined in (4). In (6), we keep the loss general as we can use different losses depending on the task.

For each experiment, we fine-tuned the step size α. The number of epochs T was chosen from $\{1, 5, 15, 50, 100\}$ and the number of validation prompts N_{v} was chosen from $\{1, 5, 10, 30, 100, 300\}$ (for details, see Appendix B). If the validation set is provided for a dataset, we utilize it as is. However, if the validation set is not available, we create one by randomly selecting a subset from the training set. As baselines, we used the vanilla ICL method (Brown et al., 2020) that does not use any calibration (NoC) and contextual calibration (ConC) (Zhao et al., 2021). The results of ConC were replicated using the released code[5]. Both ConC and our approach, LinC, utilize an affine transformation. However, the crucial distinction lies in the fact that **LinC acquires the transformation parameters through the learning process with a few additional samples** (following the same format as instruction-tuning). In contrast, ConC **does not engage in learning** and opts for a pre-defined initialization instead (for details, see Appendix C). The demonstrations' labels were not artificially balanced. Moreover, we observed that our method performs well with any set of k-shot demonstrations used in the validation prompts, but utilizing a different set can lead to improved performance, thus we conducted experiments with different random seeds to obtain a better set of the validation demonstrations.

5.2 Simulation Results

LinC enhances both the average and minimum accuracy. Table 4 shows the results on GPT-2-XL, GPT-J and Llama-2, respectively. LinC consistently outperforms baselines in almost all experiments, demonstrating its strong generalization ability across different model sizes and few-shot settings. We observe that, on average, LinC achieves up to 21% improvement as compared to the vanilla ICL baseline and 14% improvement as compared to contextual calibration for 0-shot learning on GPT-J. Moreover, in certain cases, LinC can deliver a significant boost in performance, up to 50% absolute improvement, as observed in the GPT-J 0-shot experiment on the DBPedia dataset. LinC demonstrates significant performance gain on some datasets, such as TREC, while moderate improvement is observed for other datasets, such as SST-2. Additionally, the performance improvement of LinC is more prominent on GPT-J than on GPT-2-XL and Llama-2. We also compare our results with prototypical calibration (Han et al., 2023), despite the fact that they use orders of magnitude larger number of samples and thus fall outside the few-shot learning regime.

Table 8 in Appendix B shows that LinC outperforms prototypical calibration in 16 out of 28 cases while using fewer number of samples (see Table 9). When using the same number of samples, LinC outperforms prototypical calibration in 20 of 28 cases. LinC's exceptional ability to generalize effectively could be supported by a recent discovery indicating that ICL in a conventional Transformer block is equivalent to adjusting the output layer using linear modeling of meta-learned deep data

[5] www.github.com/tonyzhaozh/few-shot-learning

Model	Shots	Method	SST-2	SST-5	AGNews	TREC	DBpedia	RTE	Subj	Avg
GPT-2-XL 1.5B	0-shot	NoC	$64.5_{0.0}$	$33.7_{0.0}$	$44.3_{0.0}$	$28.7_{0.0}$	$58.7_{0.0}$	$48.0_{0.0}$	$56.7_{0.0}$	47.8
		ConC	$70.9_{0.0}$	$20.3_{0.0}$	$65.3_{0.0}$	$41.7_{0.0}$	$50.0_{0.0}$	$50.5_{0.0}$	$73.0_{0.0}$	53.1
		LinC	$\mathbf{71.6}_{0.0}$	$\mathbf{41.3}_{0.0}$	$\mathbf{65.7}_{0.0}$	$\mathbf{42.0}_{0.0}$	$\mathbf{73.0}_{0.0}$	$\mathbf{54.5}_{0.0}$	$\mathbf{73.3}_{0.0}$	**60.2**
	1-shot	NoC	$59.7_{13.2}$	$28.3_{9.7}$	$39.6_{10.3}$	$27.1_{5.9}$	$40.5_{15.9}$	$53.4_{1.0}$	$54.6_{8.8}$	43.3
		ConC	$76.7_{1.8}$	$31.1_{4.8}$	$63.9_{3.3}$	$40.5_{3.1}$	$62.3_{7.5}$	$52.5_{1.7}$	$61.3_{7.4}$	55.5
		LinC	$\mathbf{83.2}_{16.9}$	$\mathbf{40.2}_{3.7}$	$\mathbf{64.8}_{6.3}$	$\mathbf{42.9}_{5.1}$	$\mathbf{63.1}_{7.4}$	$\mathbf{54.4}_{1.9}$	$\mathbf{73.7}_{5.5}$	**60.3**
	4-shot	NoC	$66.3_{13.1}$	$34.1_{4.8}$	$40.4_{14.1}$	$24.0_{6.3}$	$66.7_{10.2}$	$52.2_{3.2}$	$52.8_{4.9}$	48.1
		ConC	$78.9_{8.4}$	$34.4_{6.8}$	$60.4_{6.8}$	$41.3_{4.7}$	$72.1_{4.8}$	$53.0_{0.9}$	$67.8_{8.9}$	58.3
		LinC	$\mathbf{87.1}_{1.9}$	$\mathbf{39.1}_{5.6}$	$\mathbf{69.0}_{6.5}$	$\mathbf{46.0}_{3.2}$	$\mathbf{73.1}_{4.8}$	$\mathbf{53.6}_{0.6}$	$\mathbf{73.6}_{10.8}$	**63.1**
	8-shot	NoC	$57.2_{8.0}$	$31.8_{9.3}$	$41.4_{5.8}$	$31.8_{8.1}$	$59.0_{16.4}$	$52.5_{0.9}$	$56.6_{10.7}$	47.2
		ConC	$73.7_{10.5}$	$28.2_{5.4}$	$57.9_{12.2}$	$45.4_{1.7}$	$71.8_{5.7}$	$\mathbf{53.4}_{1.1}$	$68.1_{9.0}$	56.9
		LinC	$\mathbf{79.1}_{10.0}$	$\mathbf{38.1}_{9.6}$	$\mathbf{63.3}_{7.2}$	$\mathbf{48.1}_{6.1}$	$\mathbf{71.9}_{5.5}$	$53.2_{0.9}$	$\mathbf{76.9}_{3.9}$	**61.5**
GPT-J 6B	0-shot	NoC	$66.3_{0.0}$	$33.7_{0.0}$	$36.0_{0.0}$	$24.7_{0.0}$	$19.7_{0.0}$	$55.6_{0.0}$	$65.7_{0.0}$	43.1
		ConC	$58.0_{0.0}$	$40.7_{0.0}$	$56.0_{0.0}$	$40.0_{0.0}$	$48.0_{0.0}$	$52.4_{0.0}$	$58.7_{0.0}$	50.5
		LinC	$\mathbf{74.3}_{0.0}$	$\mathbf{46.0}_{0.0}$	$\mathbf{64.3}_{0.0}$	$\mathbf{70.7}_{0.0}$	$\mathbf{69.3}_{0.0}$	$\mathbf{56.7}_{0.0}$	$\mathbf{70.7}_{0.0}$	**64.6**
	1-shot	NoC	$67.3_{6.7}$	$35.3_{3.5}$	$65.3_{14.3}$	$40.3_{8.8}$	$64.9_{16.6}$	$50.7_{3.7}$	$65.1_{9.9}$	55.6
		ConC	$88.3_{1.7}$	$46.9_{3.1}$	$74.9_{5.8}$	$62.6_{4.7}$	$80.2_{3.3}$	$53.4_{1.2}$	$59.1_{2.2}$	66.5
		LinC	$\mathbf{88.5}_{1.7}$	$\mathbf{50.4}_{1.3}$	$\mathbf{81.7}_{4.6}$	$\mathbf{63.0}_{4.4}$	$\mathbf{82.6}_{3.7}$	$\mathbf{56.0}_{1.8}$	$\mathbf{69.9}_{7.5}$	**70.3**
	4-shot	NoC	$88.9_{3.3}$	$46.3_{3.2}$	$72.3_{6.1}$	$37.7_{4.2}$	$82.1_{14.4}$	$55.0_{7.1}$	$57.4_{6.9}$	62.8
		ConC	$92.8_{3.1}$	$49.4_{4.8}$	$75.2_{4.1}$	$46.0_{5.1}$	$88.9_{4.9}$	$56.0_{1.9}$	$65.3_{11.8}$	67.7
		LinC	$\mathbf{94.9}_{0.9}$	$\mathbf{51.1}_{3.7}$	$\mathbf{77.9}_{5.5}$	$\mathbf{65.3}_{1.5}$	$\mathbf{89.4}_{5.4}$	$\mathbf{58.4}_{3.7}$	$\mathbf{68.4}_{11.2}$	**72.2**
	8-shot	NoC	$91.8_{6.0}$	$44.5_{4.3}$	$76.8_{9.9}$	$43.5_{6.3}$	$88.6_{3.2}$	$60.3_{2.8}$	$82.2_{5.7}$	69.7
		ConC	$93.8_{1.8}$	$44.4_{4.4}$	$78.5_{5.7}$	$52.3_{8.3}$	$\mathbf{90.8}_{2.5}$	$58.8_{4.7}$	$81.3_{6.2}$	71.4
		LinC	$\mathbf{94.8}_{1.2}$	$\mathbf{51.3}_{0.8}$	$\mathbf{83.7}_{1.8}$	$\mathbf{67.3}_{3.8}$	$90.1_{3.9}$	$\mathbf{63.2}_{4.2}$	$\mathbf{84.7}_{4.9}$	**76.4**
Llama-2 13B	0-shot	NoC	$56.3_{0.0}$	$34.0_{0.0}$	$73.3_{0.0}$	$48.7_{0.0}$	$54.7_{0.0}$	$66.1_{0.0}$	$47.3_{0.0}$	54.3
		ConC	$69.3_{0.0}$	$33.3_{0.0}$	$72.3_{0.0}$	$71.3_{0.0}$	$75.0_{0.0}$	$67.5_{0.0}$	$47.0_{0.0}$	62.2
		LinC	$\mathbf{75.3}_{0.0}$	$\mathbf{47.3}_{0.0}$	$\mathbf{85.7}_{0.0}$	$\mathbf{75.0}_{0.0}$	$\mathbf{90.7}_{0.0}$	$\mathbf{69.0}_{0.0}$	$\mathbf{48.3}_{0.0}$	**70.2**
	1-shot	NoC	$73.9_{12.8}$	$43.9_{3.5}$	$81.5_{2.7}$	$67.0_{7.2}$	$92.3_{1.7}$	$70.3_{3.5}$	$50.5_{3.9}$	68.5
		ConC	$\mathbf{94.1}_{1.7}$	$42.4_{3.9}$	$80.6_{1.9}$	$76.2_{2.5}$	$92.3_{1.2}$	$60.9_{7.7}$	$53.2_{12.2}$	71.4
		LinC	$\mathbf{94.1}_{1.7}$	$\mathbf{51.0}_{1.6}$	$\mathbf{84.1}_{1.7}$	$\mathbf{77.3}_{2.6}$	$\mathbf{93.6}_{1.6}$	$\mathbf{75.9}_{2.6}$	$\mathbf{55.6}_{10.2}$	**75.9**
	4-shot	NoC	$92.9_{2.6}$	$48.7_{4.1}$	$82.7_{3.7}$	$62.5_{14.7}$	$94.2_{1.0}$	$68.8_{8.7}$	$72.0_{11.8}$	74.5
		ConC	$97.4_{0.4}$	$44.9_{5.2}$	$80.5_{2.3}$	$75.3_{7.2}$	$94.8_{1.2}$	$75.1_{2.0}$	$72.5_{9.8}$	77.2
		LinC	$\mathbf{97.5}_{0.5}$	$\mathbf{53.0}_{0.9}$	$\mathbf{86.0}_{1.2}$	$\mathbf{75.7}_{2.1}$	$\mathbf{95.3}_{0.9}$	$\mathbf{77.0}_{0.9}$	$\mathbf{78.7}_{9.6}$	**80.5**
	8-shot	NoC	$92.2_{3.1}$	$49.1_{7.3}$	$87.0_{0.5}$	$77.3_{4.7}$	$94.9_{1.7}$	$72.9_{5.6}$	$\mathbf{82.7}_{7.4}$	79.4
		ConC	$96.7_{0.5}$	$47.1_{4.4}$	$83.8_{1.6}$	$79.1_{3.7}$	$94.6_{1.9}$	$75.3_{3.0}$	$82.5_{5.5}$	79.9
		LinC	$\mathbf{97.0}_{0.2}$	$\mathbf{51.4}_{6.1}$	$\mathbf{87.1}_{0.6}$	$\mathbf{79.7}_{3.9}$	$\mathbf{95.0}_{1.4}$	$\mathbf{76.4}_{0.6}$	$82.1_{3.1}$	**81.2**

Table 4: Comparisons among the conventional approach (NoC; (Brown et al., 2020)), ConC (Zhao et al., 2021) and LinC (*Ours*) on GPT-2, GPT-J and Llama-2. We report the mean and the standard deviation of test accuracy across different choices of the test demonstrations (the prompt is fixed). We also report the average performance across seven datasets.

representations in few-shot settings, and LinC can be seen as explicitly adjusting this output layer with the additional samples (Von Oswald et al., 2023).

We also performed an experiment to show the impact of the model sizes (e.g., 7B vs 13B) on performance within the same model family (e.g., Llama-2); see Table 11. The results demonstrate that LinC consistently improves performance regardless of the model size.

LinC reduces variance across demonstrations. Figure 15 shows the difference in standard deviation between the calibration methods and the NoC baseline for all GPT-J experiments in Table 4. In most cases, LinC significantly reduces variance while only slightly increasing it in the remaining cases. Moreover, on average, LinC achieves a greater reduction in standard deviation than ConC, indicating that the predictions made by LinC are more consistent and reliable.

LinC improves the Expected Calibration Error (ECE). To further evaluate LinC's model calibration, we use the ECE metric (Pakdaman Naeini et al., 2015) that is widely-used to quantify the alignment between predicted and actual probabilities. Table 2 shows the results on GPT-J, 0-shot setting (see Appendix B for other settings). In most instances, LinC demonstrates the lowest ECE, and in others, it ranks as the second-

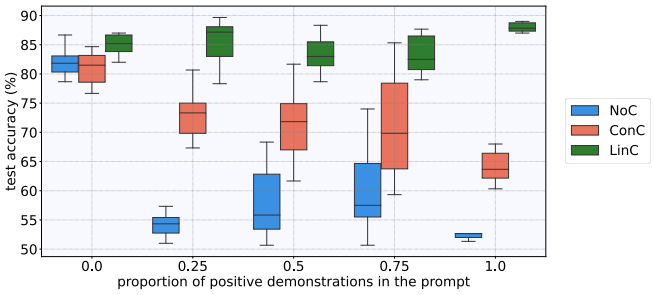

Figure 6: Comparison under varying label proportions and permutations of demonstrations; each box represents the test accuracy obtained from eight random permutations.

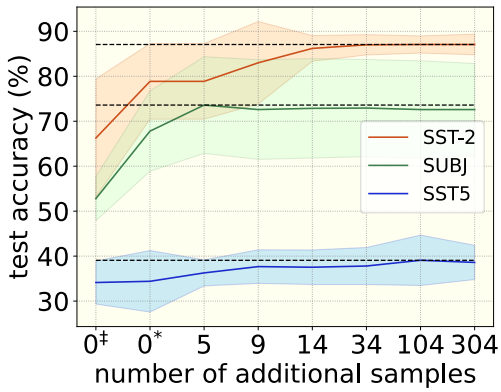

Figure 7: Performance across varying validation sizes; ‡ denotes NoC, and * denotes ConC. Black dotted line marks the maximum accuracy.

best method, with only a minimal difference from the best method.

LinC improves accuracy and reduces variance across varying prompt templates. Next we keep the set of demonstrations fixed and vary the prompt format. We use six different prompt formats and label spaces for SST-2 dataset (for details, refer to Appendix A, Table 5) on GPT-2-XL under 4-shot setting. From Figure 5, we observe that while ConC generally enhances the average accuracy, it results in high variance. In contrast, LinC exhibits a considerable enhancement in accuracy with much lower variance, demonstrating its effectiveness in improving the model's performance across various prompt templates.

LinC is robust to class imbalance and permutations of demonstrations. Prior works (Zhao et al., 2021; Lu et al., 2022) have shown that the order in which the demonstrations are set in the prompt can significantly affect the performance. We evaluated the performance of our method on five 8-shot prompts for SST-2 on GPT-2-XL, each with varying class proportions, and measured the accuracy across eight random permutations for each proportion. The demonstrations in the test prompt are kept the same for each proportion for both baselines and our method. Figure 6 illustrates that vanilla ICL (NoC) can achieve satisfactory performance with certain proportions and permutations, but performs much worse in most cases. This result is in line with the observations made in previous works (Lu et al., 2022). Moreover, ConC achieves superior test accuracy on average when compared to NoC but the performance is volatile across different permutations. LinC stands out as the most effective model and displays low variance across different permutations, suggesting that it is robust to varying class proportions and permutations.

LinC needs very few additional samples for improving ICL. Figure 7 investigates the effect of additionally available validation samples on the performance of three different datasets on GPT-2-XL under the 4-shot setting. The number of extra samples employed in the validation prompts (i.e. $N_{\mathrm{v}}+k$) is plotted along the x-axis. We observe that performance can be greatly improved by increasing the number of validation samples within a certain small range. However, further increasing the number of samples does not yield any additional improvement. LinC demonstrates high sample efficiency, achieving maximum accuracy with less than 30 additional samples on most datasets. Remarkably, some datasets require only five additional samples (e.g., Subj) to achieve maximum accuracy.

LinC also improves performance on non-language tasks. We also evaluated the performance of our method on non-language classification tasks using seven real tabular datasets in OpenML (Vanschoren et al., 2014) following the settings in (Dinh et al., 2022). Table 3 demonstrates the performance improvement of LinC over raw ICL and ConC across all tabular datasets, regardless of the number of classes (C) and data features (D). In contrast, ConC sometimes falls short in improving performance, especially in datasets with fewer classes and features, and has worse performance than raw ICL in datasets such as Hamster, Customers, and Breast.

LinC improves PEFT methods in the low-data low-compute regime. We evaluated the performance of LinC on two popular PEFT methods: 1) SPT (Lester et al., 2021) and 2) LoRA (Hu et al., 2022) (details in Appendix B.1). Tables 12 and 13 demonstrate that LinC consistently boosts both methods during fine-tuning across different sample sizes. However, performance improvement is most prominent when data and compute resources are limited. For example, for a sample size of 120 (out of 120k available training samples), LinC yields an accuracy improvement of +38.7% for

SPT and +29.3% for LoRA. This makes LinC especially viable in scenarios with constrained compute and a scarcity of data.

6 Related Work

Understanding ICL. Some recent works focus on explaining the working of ICL. For example, (Xie et al., 2022) suggested that ICL is an implicit Bayesian inference and proved it through a synthetic dataset with a mixture of hidden Markov models in pretraining. (Garg et al., 2022) showed that Transformers can learn effective learning algorithms for unseen linear functions based on demonstration samples and achieve comparable error to least squares estimator in ICL models. (Dai et al., 2023) explain that ICL can be considered as implicit finetuning, where LLM generates meta-gradients from in-context demonstrations to adjust the model's behavior. (Li et al., 2023) framed ICL as an algorithm learning problem and demonstrated that Transformers can effectively implement a function class through implicit empirical risk minimization based on demonstrations. (Von Oswald et al., 2023) showed self-attention-only Transformers trained on simple regression tasks exhibit significant similarity to models learned by gradient descent, revealing how trained Transformers execute gradient descent during their forward pass. (Chan et al., 2022) showed that ICL's performance is influenced by the distributional properties of training data, with improved performance observed when training data consists of clustered examples and sufficient rare classes. ICL is also inherently connected to multi-task learning (Caruana, 1997; Zhang and Yang, 2022) and meta-learning (Finn et al., 2017; Abbas et al., 2022; Chen et al., 2022a; Kang et al., 2023; Ji et al., 2022). But a key difference between ICL and those methods is that, in ICL, adaptation to a new task is done implicitly through input prompt not running explicit gradient-based optimization.

Enhancing ICL. To enhance the performance of ICL, meta-learning has been introduced by (Chen et al., 2022b; Min et al., 2022) to improve the adaptation of LLMs to ICL. (Wei et al., 2022b) suggest augmenting the demonstrations by incorporating human-aided reasoning steps, leading to an improvement in performance on a range of arithmetic and reasoning tasks. (Wei et al., 2022a; Sanh et al., 2022) suggest instruction tuning as a method to further pre-train the language model with a variety of downstream tasks in a shared prompting format. Several works focus on finding good in-context demonstrations to improve ICL performance (Liu et al., 2022; Levy et al., 2022). (Wang et al., 2023) investigates ICL using a Bayesian approach, and proposes an algorithm for selecting optimal demonstrations, showing empirical improvement compared to a random selection baseline. (Si et al., 2023) studies model reliability using four different facets. (Han et al., 2023) proposes estimating prototypical clusters for all classes, mapping each cluster to the corresponding label, and calibrating the test predictions by their most likely cluster. It has been found in (Jiang et al., 2023) that the sensitivity of the language model stems from the label shift of the model in the data distribution, where the model exhibits a shift in the label marginal while maintaining a strong label conditional. Their solution involves a generative calibration approach, adjusting the label marginal through Monte-Carlo sampling over the in-context model to calibrate the predictive distribution. (Zhou et al., 2023) introduce Batch Calibration (BC), a zero-shot calibration method aimed at reducing bias from the batch. Closely related to our work is (Zhao et al., 2021) which observed that the few-shot performance of language models is not consistent across different in-context settings. Additionally, (Zhao et al., 2021) notes that language models tend to predict certain labels due to bias or demonstration permutations. However, the proposed calibration method of selecting content-free test inputs in (Zhao et al., 2021) cannot accurately reflect the bias of models, which can result in sub-optimal performance. In contrast, our LinC approach achieves calibration of bias by accurately optimizing calibration parameters at the expense of minimal compute and data.

7 Conclusions

This paper investigates in-context learning (ICL), which relies on GPT-like models to generate outputs based on in-context demonstrations. Our findings reveal that ICL predictions may be unreliable when evaluated using Shannon entropy. To overcome these limitations, we propose the linear probe calibration (LinC) method, which significantly boosts the test performance of GPT models on various benchmark datasets with only a minimal number of additional samples. LinC's ability to reduce ECE and variance across different sets of demonstrations, and maintain robustness towards varying label proportions, prompt templates, and demonstration permutations implies that the predictions made by the LLM were more reliable, supporting our original conclusion from the Shannon entropy metric analysis. We believe that these findings carry important implications for future research and the development of more reliable and effective natural language processing models.

Acknowledgments

The work of T. Chen and M. Abbas was supported by the Rensselaer-IBM AI Research Collaboration (`http://airc.rpi.edu`), part of the IBM AI Horizons Network (`http://ibm.biz/AIHorizons`).

References

Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-maml: Sharpness-aware model-agnostic meta learning. In *Proc. of International Conference on Machine Learning*, Baltimore, MD, 2022.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*, 2022.

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. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

Stephanie Chan, Adam Santoro, Andrew Lampinen, Jane Wang, Aaditya Singh, Pierre Richemond, James McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 18878–18891. Curran Associates, Inc., 2022.

Lisha Chen, Songtao Lu, and Tianyi Chen. Understanding benign overfitting in gradient-based meta learning. In *Advances in Neural Information Processing Systems*, pages 19887–19899, 2022a.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland, May 2022b.

Leda Cosmides and John Tooby. Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty. *Cognition*, 58(1):1–73, 1996. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(95)00664-8. URL `https://www.sciencedirect.com/science/article/pii/0010027795006648`.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *First PASCAL Machine Learning Challenges Workshop*, pages 177–190, 2005.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics*, pages 4005–4019, Toronto, Canada, July 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. In *Advances in Neural Information Processing Systems*, pages 11763–11784, 2022.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. of International Conference on Machine Learning*, 2017.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30583–30598. Curran Associates, Inc., 2022.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

Umang Gupta, Jwala Dhamala, Varun Kumar, Apurv Verma, Yada Pruksachatkun, Satyapriya Krishna, Rahul Gupta, Kai-Wei Chang, Greg Ver Steeg, and Aram Galstyan. Mitigating gender bias in distilled language models via counterfactual role reversal. *arXiv preprint arXiv:2203.12574*, 2022.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. Prototypical calibration for few-shot learning of language models. In *Proc. of International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=nUsP9lFADUF`.

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.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Theoretical convergence of multi-step model-agnostic meta-learning. *J. Mach. Learn. Res.*, 23(1), jan 2022. ISSN 1532-4435.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association : JAMIA*, 19:263–74, 03 2012. doi: 10.1136/amiajnl-2011-000291.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. Generative calibration for in-context learning. *arXiv preprint arXiv:2310.10266*, 2023.

Suhyun Kang, Duhun Hwang, Moonjung Eo, Taesup Kim, and Wonjong Rhee. Meta-learning with a geometry-adaptive preconditioner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16080–16090, June 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=UYneFzXSJWh.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 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.

Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800*, 2022.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19565–19594. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/li23l.html.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland, May 2022.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 8086–8098, Dublin, Ireland, May 2022.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.201. URL https://aclanthology.org/2022.naacl-main.201.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nzpLWnVAyah.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 2901–2907, April 2015.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd*

Annual Meeting of the Association for Computational Linguistics (ACL-04), pages 271–278, Barcelona, Spain, July 2004. doi: 10.3115/1218955.1218990. URL https://aclanthology.org/P04-1035.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryenvpEKDr.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3): 379–423, 1948.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=98p5x51L5af.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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*, pages 1631–1642, 2013.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 819–862, Dublin, Ireland, May 2022.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

Ellen M. Voorhees and Dawn M. Tice. Building a question answering test collection. page 200–207, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132263. doi: 10.1145/345508.345577. URL https://doi.org/10.1145/345508.345577.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran

Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Ben Wang and Aran Komatsuzaki. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. In *Workshop on Efficient Systems for Foundation Models @ ICML2023*, 2023. URL https://openreview.net/forum?id=HCkI1b6ksc.

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*, 2022a. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022b.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Benfeng Xu, Quan Wang, Zhendong Mao, Yajuan Lyu, Qiaoqiao She, and Yongdong Zhang. kNN prompting: Beyond-context learning with calibration-free nearest neighbor inference. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fe2S7736sNS.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022. doi: 10.1109/TKDE.2021.3070203.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *arXiv preprint arXiv:2309.17249*, 2023.

Supplementary Material for "Enhancing In-context Learning via Linear Probe Calibration"

A Prompt Templates

Format #	Prompt Template	Label Space
1	Review: Perhaps the best sports movie I have ever seen. Sentiment: Positive Review: This pathetic junk is barely an hour long. Sentiment:	Positive, Negative
2	Input: Perhaps the best sports movie I have ever seen. Prediction: Positive Input: This pathetic junk is barely an hour long. Prediction:	Positive, Negative
3	Review: Perhaps the best sports movie I have ever seen. Sentiment: good Review: This pathetic junk is barely an hour long. Sentiment:	good, bad
4	Input: Perhaps the best sports movie I have ever seen. Prediction: good Input: This pathetic junk is barely an hour long. Prediction:	good, bad
5	Perhaps the best sports movie I have ever seen. *My overall feeling was that the movie was* good This pathetic junk is barely an hour long. *My overall feeling was that the movie was* ____	good, bad
6	Review: Perhaps the best sports movie I have ever seen. Question: Is the sentiment of the above review Positive or Negative? Answer: Positive Review: This pathetic junk is barely an hour long. Question: Is the sentiment of the above review Positive or Negative? Answer:	Positive, Negative

Table 5: A list of different prompt templates that were used to investigate the impact of templates on SST-2. For brevity, here we show only one demonstration.

Dataset	Prompt Template	Label Space
All OpenML datasets	When we have x1=r.x1, x2=r.x2, . . . , xK=r.xD, what should be y? ### y=r.y @@@	0,. . . ,C

Table 6: Prompt template used for non-language classification tasks using the real tabular datasets in OpenML (Vanschoren et al., 2014); for a sample r, D denotes the number of features; Following (Dinh et al., 2022), we follow OpenAI's recommendation by using "###" for separating questions and answers, and "@@@" to indicate the end of answer.

Dataset	Prompt Template	Label Space
SST-2	Review: Perhaps the best sports movie I have ever seen. Sentiment: Positive Review: This pathetic junk is barely an hour long. Sentiment:	Positive, Negative
AGNews	Classify the news articles into the categories of World, Sports, Business, and Technology. Article: UK lender Barclays says it is in talks with South Africa's Absa about buying a majority stake in the bank. Answer: Business Article: New music sharing network allows users to amass points by referring buyers. Answer:	World, Sports, Business, Technology
TREC	Classify the questions based on whether their answer type is a Number, Location, Person, Description, Entity, or Abbreviation. Question: What does the abbreviation AIDS stand for? Answer Type: Abbreviation Question: What country do the Galapagos Islands belong to? Answer Type:	Number, Location, Person, Description, Entity, Abbreviation
DBPedia	Classify the documents based on whether they are about a Company, School, Artist, Athlete, Politician, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, or Book. Article: Hoodlum & Son is a 2003 comedy-crime film. Answer: Film Article: Nachan Main Audhay Naal is the seventh album of Pakistani pop and bhangra singer Abrar-ul-Haq. It was released on March 2007. Answer:	Company, School, Artist, Athlete, Politician, Transportation, Building, Nature, Village, Animal, Plant, Album, Film, Book
SST-5	Review: The film is bright and flashy in all the right ways. Sentiment: great Review: The film never finds its tone and several scenes run too long. Sentiment:	terrible, bad, okay, good, great
Subj	Input: All social structures break down and a new world order emerges from the heart of the desert. Type: objective Input: A zombie movie in every sense of the word – mindless, lifeless, meandering, loud , painful, obnoxious. Type:	objective, subjective
RTE	Experts say that Mr. Abbas will need that big win to show that he has the support of most Palestinian people in order to push through his aims of peace talks with Israel. question: Analysts had said that Mr. Abbas needed a large margin of victory in order to push his agenda of peace talks with Israel. True or False? answer: True The city is twinned with Glasgow, Dortmund, Pleven, and Le Mans. question: Dortmund is twinned with Glasgow. True or False? answer:	True, False

Table 7: The prompts templates used for different datasets. For brevity, here we show only one demonstration per dataset.

B Additional Experiments

In this section, we provide details of the experimental set-up and present additional results.

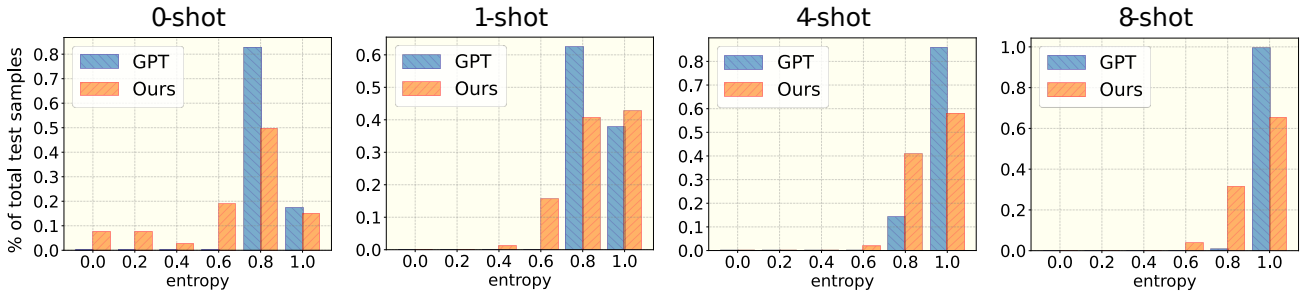

Figure 8: Shannon entropy histograms of using vanilla ICL on GPT-2-XL (1.5B) vs our method on Subj (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

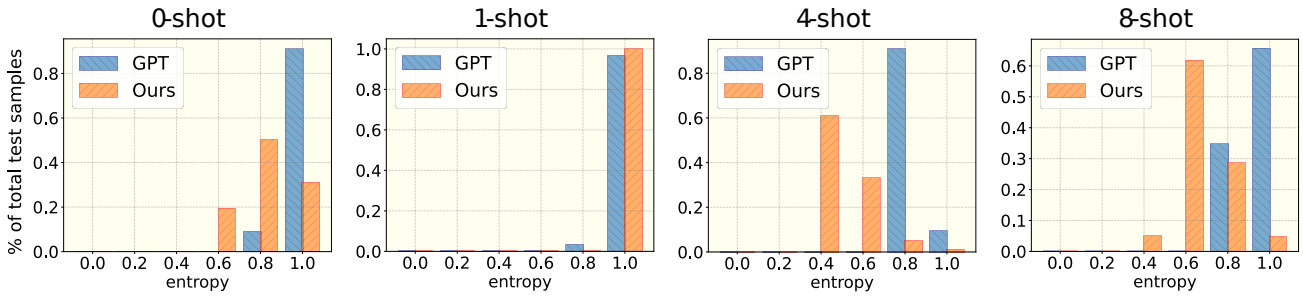

Figure 9: Shannon entropy histograms of using vanilla ICL on GPT-2-XL (1.5B) vs our method on RTE (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

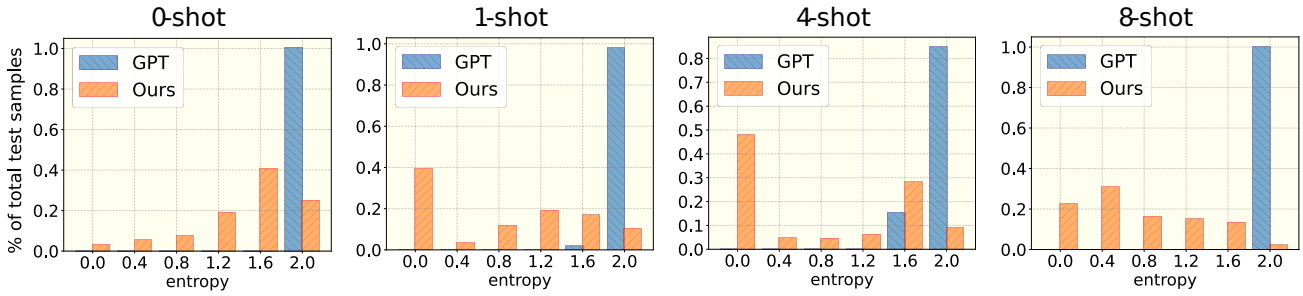

Figure 10: Shannon entropy histograms of using vanilla ICL on GPT-2-XL (1.5B) vs our method on AGNews (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

Hyperparameter search. After finding N_v and T via random search using the sets defined in Section 5.1, we narrowly fine-tune the learning rate α for each experiment from the range $[1e-5, 2e1]$. Therefore, in each experiment, the value of α might vary, and to ensure transparency and reproducibility, we have compiled some hyperparameter sets in the examples_ssh folder within the provided code.

Furthermore, given that the calibration network constitutes a basic linear convex problem, determining an appropriate learning rate presents no issue, as the fixed rate can readily be substituted with a schedule that gradually reduces the learning rate as training advances, based on specified criteria (e.g., a linear scheduler that gradually diminishes the rate for each parameter group by applying a small multiplicative factor until a predetermined epoch count is attained).

Where is the affine transformation applied? The classification tasks we considered apply the affine transformation to the set of probabilities that are associated with each class in the label space i.e. task-specific

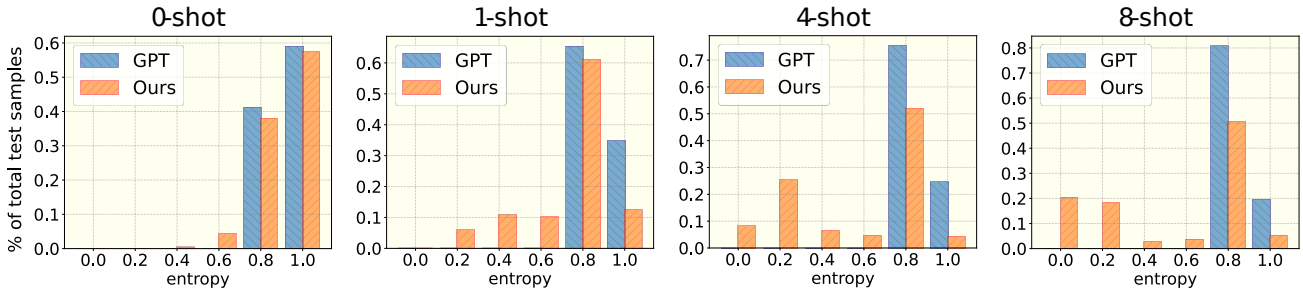

Figure 11: Shannon entropy histograms of using vanilla ICL on Llama-2 (13B) vs our method on SST-2 (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

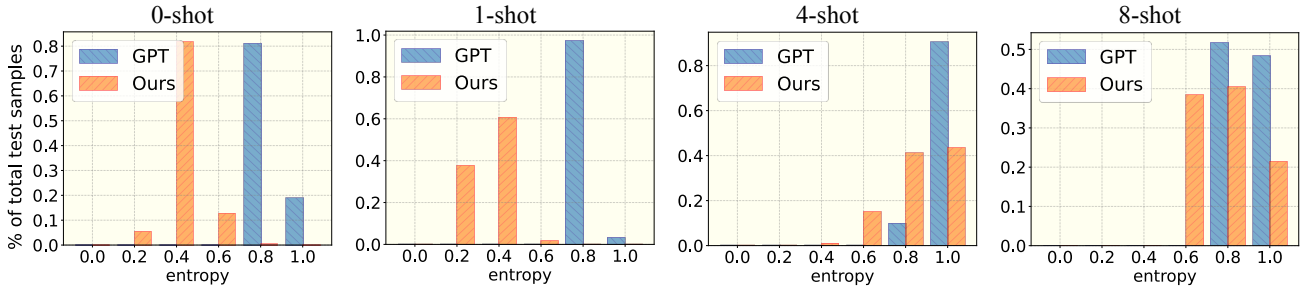

Figure 12: Shannon entropy histograms of using vanilla ICL on Llama-2 (13B) vs our method on Subj (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

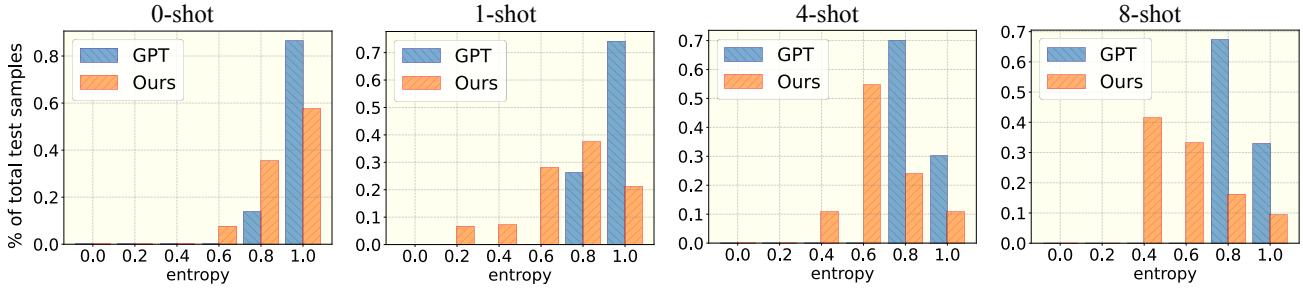

Figure 13: Shannon entropy histograms of using vanilla ICL on Llama-2 (13B) vs our method on RTE (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

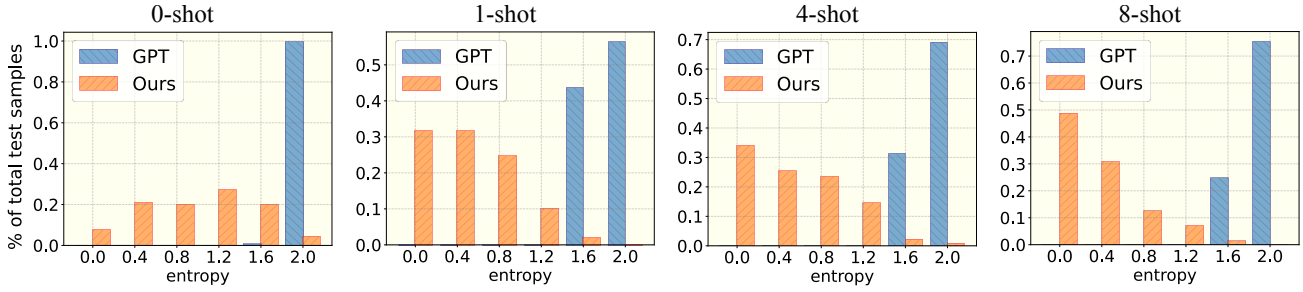

Figure 14: Shannon entropy histograms of using vanilla ICL on Llama-2 (13B) vs our method on AGNews (higher entropy implies higher uncertainty); we use logarithmic base two. Refer to Section 2 for a detailed explanation.

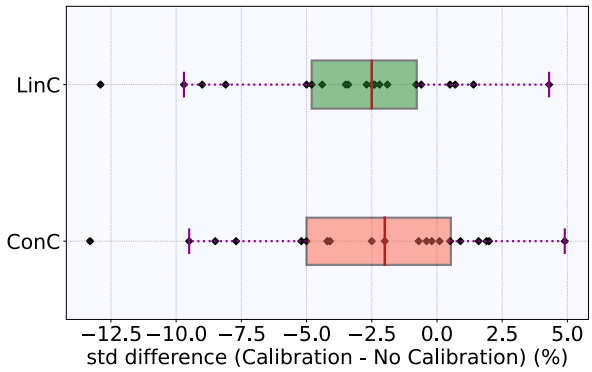

Figure 15: LinC diminishes the standard deviation of accuracy across different demonstrations. The difference in standard deviation between the calibration methods and the unadjusted baseline (NoC) from Table 4 is plotted.

tokens. In other words, after we get the logits which are in the dimension (batch_size, seq_length, vocab_size), we sample the tokens that correspond to the classes in the label space to a get a probability vector of dimension (batch_size, num_classes). We then apply the affine transformation on this probability vector.

Expected Calibration Error (ECE) (Pakdaman Naeini et al., 2015). The ECE computes the Expected Calibration Error across the bins in the following manner:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |o_m - e_m| \tag{8}$$

where o_m represents the accuracy i.e. the true fraction of positive instances in bin m, e_m represents the model confidence i.e. is the mean of the post-calibrated probabilities for the instances in bin m, and the fraction $\frac{B_m}{n}$ denotes the empirical probability (fraction) of all instances that fall into bin. Therefore, The ECE can be seen as evaluating the extent to which a model's estimated "probabilities" align with the actual (observed) probabilities by computing a weighted average of the absolute difference between accuracy and confidence. A model's calibration is considered better when ECE values are lower and an ECE of zero is indicative of a perfectly calibrated model.

Model	Shots	Method	SST-5	AGNews
Llama-2 7B	0-shot	NoC	$32.3_{0.0}$	$57.3_{0.0}$
		ConC	$35.0_{0.0}$	$63.3_{0.0}$
		LinC	$\mathbf{45.0}_{0.0}$	$\mathbf{82.0}_{0.0}$
	1-shot	NoC	$39.3_{9.8}$	$83.9_{2.1}$
		ConC	$43.0_{1.1}$	$82.0_{3.3}$
		LinC	$\mathbf{50.1}_{3.5}$	$\mathbf{84.3}_{2.6}$
	4-shot	NoC	$49.7_{1.9}$	$87.2_{1.1}$
		ConC	$44.3_{2.5}$	$87.2_{1.2}$
		LinC	$\mathbf{53.5}_{2.8}$	$\mathbf{87.6}_{1.1}$
	8-shot	NoC	$48.9_{5.5}$	$86.7_{0.9}$
		ConC	$48.7_{2.2}$	$87.5_{0.5}$
		LinC	$\mathbf{52.0}_{3.3}$	$\mathbf{87.6}_{0.6}$
Llama-2 13B	0-shot	NoC	$34.0_{0.0}$	$73.3_{0.0}$
		ConC	$33.0_{0.0}$	$72.3_{0.0}$
		LinC	$\mathbf{47.3}_{0.0}$	$\mathbf{85.7}_{0.0}$
	1-shot	NoC	$43.9_{3.5}$	$81.5_{2.7}$
		ConC	$42.4_{3.9}$	$80.6_{1.9}$
		LinC	$\mathbf{51.0}_{1.6}$	$\mathbf{84.1}_{1.7}$
	4-shot	NoC	$48.7_{4.1}$	$82.7_{3.7}$
		ConC	$44.9_{5.2}$	$80.5_{2.3}$
		LinC	$\mathbf{53.0}_{0.9}$	$\mathbf{86.0}_{1.2}$
	8-shot	NoC	$49.1_{7.3}$	$87.0_{0.5}$
		ConC	$47.1_{4.4}$	$83.8_{1.6}$
		LinC	$\mathbf{51.4}_{6.1}$	$\mathbf{87.1}_{0.6}$

Table 11: Performance under same model family (i.e. Llama-2) but different sizes (7B/13B).

B.1 Experiment Settings

For LoRA, we used a step size of 1.5e-6, a batch size of 16, a scaling factor of 32, and incorporated a 0.1 dropout probability within the LoRA layers. For SPT, we used a step size of 3.5e-4, a batch size of 16, specified 8 virtual tokens (i.e. the num_vir_tokens argument), and employed an initial text for prompt tuning (i.e. the prompt_tuning_init_text argument) that consisted of a list of classes separated by commas (for example, "World, Sports, Business, Technology" for AGNews). Across all our experiments, we performed fine-tuning over a total of 20 epochs.

For the non-language tasks that used tabular datasets from OpenML (Vanschoren et al., 2014), following (Dinh et al., 2022), we selected the maximum number of instances that could be accommodated within the contextual window for each dataset. Additionally, our results were reported as an average across three random seeds.

Shot	Method	SST-2	SST-5	AGNews	TREC	DBpedia	RTE	Subj	Avg
				GPT-J 6B					
0-shot	No Calibration	$66.3_{0.0}$	$33.7_{0.0}$	$36.0_{0.0}$	$24.7_{0.0}$	$19.7_{0.0}$	$55.6_{0.0}$	$65.7_{0.0}$	43.1
	Contextual Calibration	$58.0_{0.0}$	$40.7_{0.0}$	$56.0_{0.0}$	$40.0_{0.0}$	$48.0_{0.0}$	$52.4_{0.0}$	$58.7_{0.0}$	50.5
	Prototypical Calibration‡	$74.2_{0.2}$	$42.1_{0.8}$	$55.1_{0.4}$	$53.4_{6.1}$	$66.1_{1.5}$	$\mathbf{57.0}_{1.0}$	$69.5_{0.2}$	59.6
	Prototypical Calibration*	$74.1_{1.1}$	$32.2_{3.8}$	$49.8_{4.3}$	$44.7_{6.7}$	$20.7_{4.7}$	$56.5_{1.5}$	$66.8_{1.8}$	49.3
	Linear Calibration	$\mathbf{74.3}_{0.0}$	$\mathbf{46.0}_{0.0}$	$\mathbf{64.3}_{0.0}$	$\mathbf{70.7}_{0.0}$	$\mathbf{69.3}_{0.0}$	$56.7_{0.0}$	$\mathbf{70.7}_{0.0}$	**64.6**
1-shot	No Calibration	$67.3_{6.7}$	$35.3_{3.5}$	$65.3_{14.3}$	$40.3_{8.8}$	$64.9_{16.6}$	$50.7_{3.7}$	$65.1_{9.9}$	55.6
	Contextual Calibration	$88.3_{1.7}$	$46.9_{3.1}$	$74.9_{5.8}$	$62.6_{4.7}$	$80.2_{3.3}$	$53.4_{1.2}$	$59.1_{2.2}$	66.5
	Prototypical Calibration‡	$\mathbf{90.8}_{1.7}$	$47.6_{2.5}$	$79.8_{5.4}$	$55.3_{6.4}$	$\mathbf{90.0}_{2.2}$	$56.7_{3.1}$	$\mathbf{77.9}_{4.8}$	**71.2**
	Prototypical Calibration*	$89.3_{1.9}$	$35.2_{8.9}$	$78.1_{7.5}$	$45.9_{9.6}$	$62.2_{4.9}$	$\mathbf{57.7}_{1.8}$	$74.9_{5.9}$	63.2
	Linear Calibration	$88.5_{1.7}$	$\mathbf{50.4}_{1.3}$	$\mathbf{81.7}_{4.6}$	$\mathbf{63.0}_{4.4}$	$82.6_{3.7}$	$56.0_{1.8}$	$69.9_{7.5}$	70.3
4-shot	No Calibration	$88.9_{3.3}$	$46.3_{3.2}$	$72.3_{6.1}$	$37.7_{4.2}$	$82.1_{14.4}$	$55.0_{7.1}$	$57.4_{6.9}$	62.8
	Contextual Calibration	$92.8_{3.1}$	$49.4_{4.8}$	$75.2_{4.1}$	$46.0_{5.1}$	$88.9_{4.9}$	$56.0_{1.9}$	$65.3_{11.8}$	67.7
	Prototypical Calibration‡	$\mathbf{95.0}_{0.4}$	$46.2_{4.6}$	$79.9_{6.6}$	$57.1_{5.3}$	$\mathbf{91.9}_{2.6}$	$\mathbf{61.2}_{2.7}$	$\mathbf{79.4}_{5.8}$	**73.0**
	Prototypical Calibration*	$94.6_{0.6}$	$47.9_{5.5}$	$\mathbf{81.5}_{2.0}$	$49.0_{10.2}$	$63.8_{4.3}$	$60.8_{3.1}$	$78.9_{8.0}$	68.1
	Linear Calibration	$94.9_{0.9}$	$\mathbf{51.1}_{3.7}$	$77.9_{5.5}$	$\mathbf{65.3}_{1.5}$	$89.4_{5.4}$	$58.4_{3.7}$	$68.4_{11.2}$	72.2
8-shot	No Calibration	$91.8_{6.0}$	$44.5_{4.3}$	$76.8_{9.9}$	$43.5_{6.3}$	$88.6_{3.2}$	$60.3_{2.8}$	$82.2_{5.7}$	69.7
	Contextual Calibration	$93.8_{1.8}$	$44.4_{4.4}$	$78.5_{5.7}$	$52.3_{8.3}$	$90.8_{2.5}$	$58.8_{4.7}$	$81.3_{6.2}$	71.4
	Prototypical Calibration‡	$94.4_{1.0}$	$47.4_{4.4}$	$\mathbf{84.2}_{1.8}$	$61.0_{7.6}$	$\mathbf{95.1}_{0.5}$	$61.7_{7.2}$	$83.6_{4.2}$	75.3
	Prototypical Calibration*	$94.1_{1.9}$	$46.8_{8.7}$	$83.7_{1.4}$	$47.5_{8.3}$	$68.5_{6.8}$	$61.4_{4.7}$	$83.8_{5.9}$	69.4
	Linear Calibration	$\mathbf{94.8}_{1.2}$	$\mathbf{51.3}_{0.8}$	$83.7_{1.8}$	$\mathbf{67.3}_{3.8}$	$90.1_{3.9}$	$\mathbf{63.2}_{4.2}$	$\mathbf{84.7}_{4.9}$	**76.4**

Table 8: Performance comparisons among the conventional approach (No Calibration; (Brown et al., 2020)), Contextual Calibration (Zhao et al., 2021), Prototypical Calibration (Han et al., 2023) and Linear Probe Calibration (*Ours*) on GPT-J 6B. ‡ denotes reporting the numbers from the original paper that utilize orders of magnitude larger number of samples (see Table 9); * denotes our reproduced results that use same number of samples as used by our method. We report the mean and the standard deviation of test accuracy across different choices of the demonstrations (the prompt template is fixed). Prototypical Calibration's 0-shot accuracy deviation results from varying estimate sets across five random seeds (Han et al., 2023). We also report the average performance across seven datasets.

C Difference from Contextual Calibration (ConC).

Both ConC and our method LinC employ an affine transformation, however, the key difference is that **LinC learns the transformation parameters using a few additional samples** (in the same format utilized by instruction-tuning). Conversely, ConC **does not learn** and instead employs a pre-defined initialization. More specifically, they create a prompt $P_{\text{cf}} = P(\text{"NA"}, (x_i, y_i)_{i=1}^{k})$ and obtain $\mathbf{p}$ for this content-free input, denoted by $\mathbf{p}_{\text{cf}}$ i.e. $\mathbf{p}_{\text{cf}} = \mathbf{p}(P_{\text{cf}})$. The parameters are set via:

$$\mathbf{A} = \mathsf{diag}(\mathbf{p}_{\text{cf}})^{-1} \quad \text{and} \quad \mathbf{b} = \mathbf{0}. \tag{9}$$

They use Equation (9) to **hard-code these parameters**. On the other hand, LinC starts with a set of calibration parameters where zero initilization is one of the possible initialization method, see Table 10. Then LinC utilizes a few additional data in the form of prompts to train the matrix $\mathbf{A}$ and vector $\mathbf{b}$ before applying the affine transformation. Note that LinC requires **orders of magnitue less data than the existing calibration methods** such as ProCa (Han et al., 2023). For more in-depth comparison results, please refer to Table 9. Remarkably, in some tasks it suffices for LinC to use **as few as only 5 additional samples** (see Figure 7).

D CO2 Emission Related to Experiments

Experiments in this paper were conducted using Amazon Web Services in region us-east-1, which has a carbon efficiency of 0.37 $\text{kgCO}_2\text{eq/kWh}$. For each experiment, the total emissions are estimated to be 0.055 kgCO_2eq, which are calculated using the MachineLearning Impact calculator in (Lacoste et al., 2019). While training or fine-tuning an LLM can result in significantly larger carbon emissions, with GPT-3 training requiring around 500,000 kgCO_2eq (Patterson et al., 2021) and GPT-2-XL estimated to emit between 100,000 and 200,000 kgCO_2eq,

Shot	Method	SST-2	SST-5	AGNews	TREC	DBpedia	RTE	Subj
				GPT-J 6B				
0-shot	Prototypical Calibration	500	2000	2000	2000	3000	1000	1000
	Linear Calibration	100	30	300	300	300	100	100
1-shot	Prototypical Calibration	500	2000	2000	2000	3000	1000	1000
	Linear Calibration	300	30	300	100	30	100	100
4-shot	Prototypical Calibration	500	2000	2000	2000	3000	1000	1000
	Linear Calibration	100	100	100	100	30	100	100
8-shot	Prototypical Calibration	500	2000	2000	2000	3000	1000	1000
	Linear Calibration	100	100	100	100	30	100	100

Table 9: Validation sample size N_{v} used for different datasets

Method	SST-2	SST-5	AGNews	RTE
		GPT-J 6B		
Zero Initialization	$74.7_{0.0}$	$43.0_{0.0}$	$65.0_{0.0}$	$56.7_{0.0}$
Random Initialization	$74.7_{0.0}$	$43.0_{0.0}$	$65.0_{0.0}$	$56.7_{0.0}$
Initialize via Eq. (9)	$74.3_{0.0}$	$46.0_{0.0}$	$64.3_{0.0}$	$56.7_{0.0}$

Table 10: Performance comparisons among different initialization methods on 0-shot setting.

our method trains low-dimensional parameters with minimal carbon emissions that are comparable to the inference stage. This makes our approach environmentally friendly in comparison to other methods.

Limitations and Future Work

While our focus remains on calibrating the model for better accuracy and reliability, it is worth discussing how to combine our framework with approaches for selecting better examples and prompt templates, such as those proposed in (Liu et al., 2022; Sorensen et al., 2022). Our work focuses on querying LLMs, which can generate content with potential ethical risks such as fairness and bias; thus, combining our framework with methods (Gupta et al., 2022) that mitigate such risks is worth further discussion. One limitation of our work is the choice of k-shot validation demonstrations used to optimize the calibration parameters can impact the quality of the learned parameters, potentially leading to suboptimal results. Besides overcoming this limitation, we aim to expand our method to other NLP tasks such as summarization, text generation, and generative question answering.

	Method	120	600	1.2k	6k
		GPT-J 6B			
	SPT	27.7	26.3	43.3	78.7
Ep#1	SPT+LinC	66.3	59.7	73.3	83.0
	Difference	+38.7	+33.3	+30.0	+4.3
	SPT	41.0	57.7	61.7	86.0
Ep#3	SPT+LinC	69.3	74.7	79.0	86.7
	Difference	+28.3	+17.0	+17.3	+0.7
	SPT	30.7	51.3	55.7	84.3
Ep#10	SPT+LinC	42.3	74.7	77.7	86.0
	Difference	+11.7	+23.3	+22.0	+1.7
	SPT	29.7	58.7	65.3	84.0
Ep#20	SPT+LinC	34.7	77.0	77.7	86.3
	Difference	+5.0	+18.3	+12.3	+2.3
	SPT	41.0	62.3	71.0	86.0
Best	SPT+LinC	69.3	80.3	81.3	86.7
	Difference	+28.3	+18.0	+10.3	+0.7
		Llama-2 13B			
	SPT	43.0	44.7	55.0	63.3
Ep#1	SPT+LinC	80.3	84.7	85.0	84.0
	Difference	+37.3	+40.0	+30.0	+20.7
	SPT	41.3	61.3	64.7	71.0
Ep#3	SPT+LinC	79.7	85.3	78.7	84.0
	Difference	+38.4	+24.0	+14.0	+13.0
	SPT	49.0	39.3	79.3	86.0
Ep#10	SPT+LinC	84.3	69.3	84.0	86.3
	Difference	+35.3	+30.0	+4.7	+0.3
	SPT	62.0	79.0	84.0	84.7
Ep#20	SPT+LinC	79.3	84.7	84.7	85.3
	Difference	+17.3	+5.7	+0.7	+0.6
	SPT	62.0	79.0	84.0	87.3
Best	SPT+LinC	79.3	84.7	84.7	87.3
	Difference	+17.3	+5.7	+0.7	0.0

Table 12: We assess the impact of Soft Prompt Tuning (SPT) on GPT-J 6B and Llama-2 13B using AGNews dataset with and without LinC in two distinct scenarios: 1) high compute and abundant data (lower-right), and 2) limited compute and sparse data (upper-left). The columns represent the number of data points employed for SPT, while the rows represent the number of training epochs.

	Method	120	600	1.2k	6k
		GPT-J 6B			
Ep#1	LoRA	36.7	26.0	72.3	86.3
	LoRA+LinC	66.0	70.7	85.0	87.0
	Difference	+29.3	+44.7	+12.7	+0.7
Ep#3	LoRA	26.3	67.0	80.0	83.0
	LoRA+LinC	66.3	85.0	84.7	88.0
	Difference	+40.0	+18.0	+4.7	+5.0
Ep#10	LoRA	41.7	76.3	78.3	85.7
	LoRA+LinC	66.0	84.3	84.0	87.3
	Difference	+24.3	+8.0	+6.0	+1.6
Ep#20	LoRA	43.3	75.0	78.3	87.7
	LoRA+LinC	69.0	83.7	82.7	88.0
	Difference	+25.7	+8.7	+4.4	+0.3
Best	LoRA	41.7	78.3	82.0	88.7
	LoRA+LinC	66.0	83.7	83.7	88.7
	Difference	+24.3	+5.4	+1.7	0.0
		Llama-2 13B			
Ep#1	LoRA	73.3	80.0	86.7	83.3
	LoRA+LinC	85.7	86.0	87.3	87.0
	Difference	+12.4	+6.0	+0.6	+3.7
Ep#3	LoRA	73.7	84.0	86.3	85.7
	LoRA+LinC	85.7	87.3	87.7	88.0
	Difference	+12.0	+3.3	+1.4	+2.3
Ep#10	LoRA	73.7	86.3	85.0	88.0
	LoRA+LinC	85.3	87.0	86.0	89.3
	Difference	+11.6	+0.7	+1.0	+1.3
Ep#20	LoRA	73.3	86.3	85.3	86.7
	LoRA+LinC	85.0	87.0	85.7	87.7
	Difference	+11.7	+0.7	+0.4	+1.0
Best	LoRA	73.7	87.0	86.7	89.0
	LoRA+LinC	85.7	88.3	87.3	89.0
	Difference	+12.0	+1.3	+0.6	0.0

Table 13: We assess the impact of Low-Rank Adaptation (LoRA) on GPT-J 6B and Llama-2 13B using AGNews dataset with and without LinC in two distinct scenarios: 1) high compute and abundant data (lower-right), and 2) limited compute and sparse data (upper-left). The columns represent the number of data points employed for LoRA, while the rows represent the number of training epochs.

Dataset	**Llama-2 (13B)**		
	ICL	**ConC**	**LinC**
SST-2	0.1766	**0.0943**	0.1263
SST-5	0.1760	0.1681	**0.1095**
AGNews	0.1294	**0.0791**	0.0939
TREC	0.2141	0.1823	**0.0782**
DBPedia	0.2000	0.1744	**0.1124**
RTE	**0.0369**	0.0467	0.0600
Subj	0.3183	0.2658	**0.1193**

Table 14: Expected Calibration Error (ECE) comparison between baselines and LinC on Llama-2 (13B) under 0-shot setting (a model with perfect calibration would exhibit an ECE of 0).