Prompt-Based Bias Calibration for Better Zero/Few-Shot Learning of Language Models

Anonymous ACL submission

Abstract

Prompt learning is susceptible to intrinsic bias present in pre-trained language models (LMs), resulting in sub-optimal performance of prompt-based zero/few-shot learning. In this work, we propose a *null-input prompting* method to calibrate intrinsic bias encoded in pre-trained LMs. Different from prior efforts that address intrinsic bias primarily for social fairness and often involve excessive computational cost, our objective is to explore enhancing LMs' performance in downstream zero/few-shot learning while emphasizing the efficiency of intrinsic bias calibration. Specifically, we leverage a diverse set of auto-selected null-meaning inputs generated from GPT-4 to probe intrinsic bias of pre-trained LMs. Utilizing the bias-reflected probability distribution, we formulate a distribution disparity loss for bias calibration, where we exclusively update bias parameters (0.1% of total parameters) of LMs towards equal probability distribution. Experimental results show that the calibration promotes an equitable starting point for LMs while preserving language modeling abilities. Across a wide range of datasets, including sentiment analysis and topic classification, our method significantly improves zero/few-shot learning performance of LMs for both in-context learning and prompt-based fine-tuning (on average 9% and 2%, respectively).

1 Introduction

The advent of GPT models (Radford et al., 2019; Brown et al., 2020) has catalyzed the transformative prompt-learning paradigm. The innovative approach of "pre-train, prompt, and predict" (Schick and Schütze, 2021a; Liu et al., 2023) facilitates fast adaptation of pre-trained language models (LMs) in learning various tasks and empowering LMs' strong zero/few-shot learning abilities (Schick and Schütze, 2021b; Gao et al., 2021).

Due to the susceptibility to bias ingrained in pre-trained LMs, prompt learning tends to make biased predictions toward some specific answers, thereby impacting the performance of prompt-based zero/few-shot learning (Zhao et al., 2021; Han et al., 2023). To mitigate this issue and improve LM performance, Zhao et al. (2021) and Holtzman et al. (2022) propose to reweigh LM output probabilities. Han et al. (2023) explores calibrating decision boundaries. While these research has demonstrated substantial improvements, they are primarily designed for in-context learning with frozen pre-trained LMs, leading to two main limitations: (1) They may be not effective in task-specific fine-tuning scenario (Jian et al., 2022). Note, however, prompt-based fine-tuning has shown performance improvements over in-context learning (Gao et al., 2021; Logan IV et al., 2022). It is particularly important for relatively small-sized LMs. (2) The intrinsic bias encoded in pre-trained LMs persists since these research focuses on *output calibration* and does not modify LMs.

To address these limitations, we investigate the potential for enhancing the performance of LMs as zero/few-shot learners in classification tasks by *calibrating intrinsic bias* of pre-trained LMs. This exploration extends to various prompt-learning scenarios: in-context learning and prompt-based fine-tuning. Prior approaches to mitigate intrinsic bias primarily focus on achieving social fairness, and often require laborious corpora augmentation and costly re-training (Huang et al., 2020; Kaneko and Bollegala, 2021; Solaiman and Dennison, 2021; Li et al., 2023a). To improve efficiency in both data generation and model updates, we propose leveraging auto-generated *null-meaning inputs* to prompt pre-trained LMs for intrinsic bias probing, and subsequently updating only *bias parameters* $\boldsymbol{B_{LM}}$ of LMs for bias calibration. Null-meaning inputs are essentially normal text devoid of meaningful content or sentiment. Unlike numerical-zero inputs, they maintain the contextual framework of prompts, ensuring the proper functioning of contex-

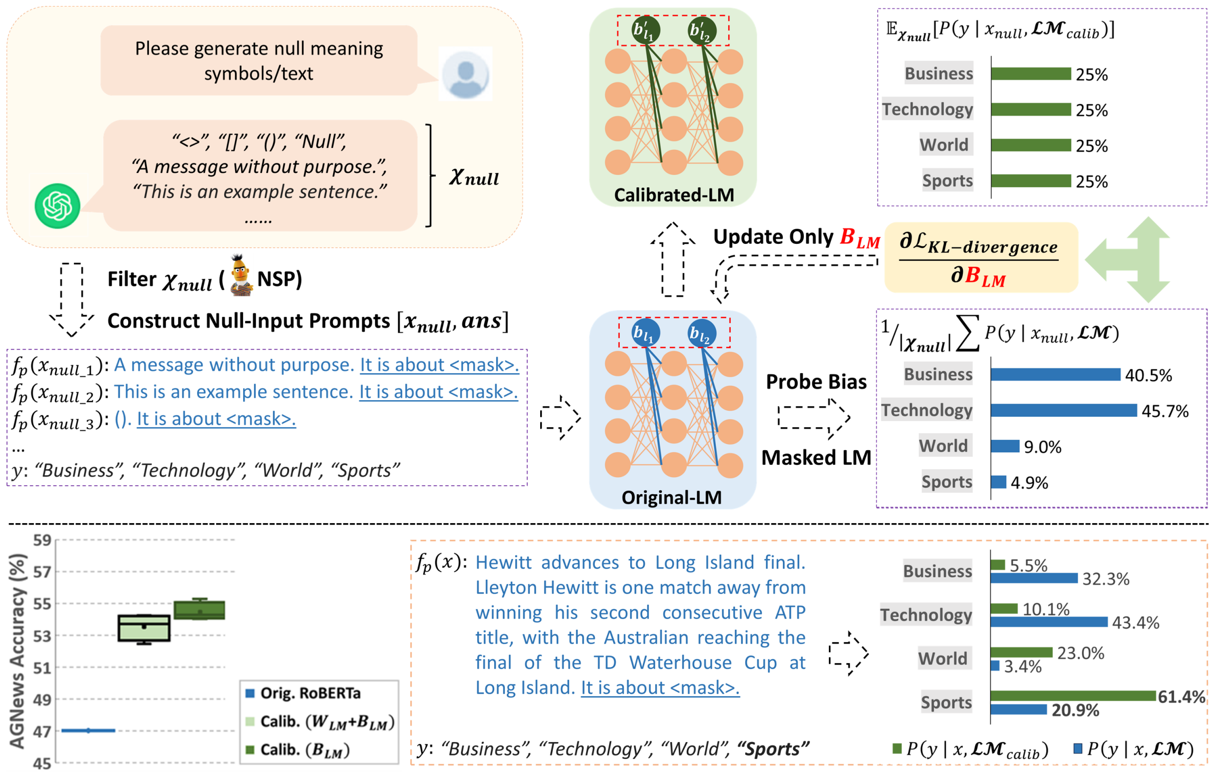

Figure 1: We demonstrate our calibration method significantly improves classification performance of pre-trained LM. **Upper**: The pipeline of proposed null-input prompting method for intrinsic bias calibration targeting AGNews task (Zhang et al., 2015). **Lower left**: Performance comparison of zero-shot in-context learning using: original LM (Orig. RoBERTa); calibrated (Calib.) LM with full model updates ($\boldsymbol{W_{LM}} + \boldsymbol{B_{LM}}$); calibrated LM with only $\boldsymbol{B_{LM}}$ updates. **Lower right**: Case study illustrating that LM makes correct prediction after intrinsic bias calibration.

tual LMs. Our motivation stems from the expectation that bias-calibrated models should produce uniform probabilities across all categories if the input in a prompt delivers null information (Zhao et al., 2021). $\boldsymbol{B_{LM}}$ functions as offsets in neural networks, and strategically updating only $\boldsymbol{B_{LM}}$ could potentially counteract intrinsic bias of pre-trained models, achieving higher efficiency (updating $\sim 0.1\%$ parameters of entire LM). The approach promotes an equitable starting point, and we expect that the light model updates preserve pre-trained models' language modeling abilities while maintaining the focus on bias calibration, ultimately making LMs better zero/few-shot learners.

The pipeline of our calibration method is illustrated in Figure 1. We use Masked LMs (RoBERTa Liu et al., 2019) for zero/few-shot learning since they generally produce competitive performance in classification tasks and their moderate size facilitates combining prompting with fine-tuning (Gao et al., 2021; Liu et al., 2023). First, we utilize GPT-4 API to automatically generate diverse null-meaning inputs $\mathcal{X}_{\text{null}}$ including symbols, words, phrases, and sentences. This generation process is downstream task-agnostic. By concatenating each null-meaning input x_{null} with an answer format *ans* aligned with the downstream task, we construct null-input prompts (similar to Zhao et al., 2021), e.g., *"An empty sentence. It is about <mask>."*. For better cohesive integration of the *"null"* information into the prompts, we additionally devise a filtering strategy to select x_{null}, to which the answer format *ans* exhibits relatively strong Next Sentence Prediction (NSP) correlation (Devlin et al., 2019). Next, we update $\boldsymbol{B_{LM}}$ with null-input prompts to calibrate intrinsic bias. Given the absence of task-relevant information in these prompts, the anticipated outcome in the parameter updating process is a convergence towards equal output probabilities for each label word. We formulate a customized Kullback–Leibler (KL) divergence loss for gradient descent on $\boldsymbol{B_{LM}}$ to minimize the distribution disparity. Finally, bias-calibrated LMs are applied in downstream prompt-based zero/few-shot learning following Gao et al. (2021).

The main contributions of our work are:

- We introduce a null-input prompting method for calibrating intrinsic bias of pre-trained

Masked LMs, aiming for better prompt-based zero/few-shot classification performance.

- Our method integrates two key aspects for efficient bias calibration: auto-construction of null-input prompts and updating only bias parameters of LMs. The calibration promotes a fair starting point for LMs while preserving language modeling abilities.
- Extensive experiments on eight classification datasets with four prompt-learning approaches show that our method significantly improves LMs' zero/few-shot performance, and outperforms output-calibration methods.

2 Related Work

Impact of intrinsic bias on downstream LM performance. Intrinsic bias in pre-trained LMs stems from imbalances present in extensive pre-training corpora. Higher frequency of specific terms in those corpora could lead to *common token bias* (Zhao et al., 2021). Additionally, frequent co-occurrence of certain terms with specific sentiment in pre-training could introduce *association bias* (Cao et al., 2022). Because of those intrinsic bias, prompt-based predictions by pre-trained LMs are prone to bias towards some specific answers, resulting in sub-optimal performance in downstream tasks (Zhao et al., 2021; Han et al., 2023).

Mitigating strategies. Research has focused on counteracting the bias solely at the output prediction stage, without modifying pre-trained LMs. For example, Zhao et al. (2021) introduces contextual calibration and Holtzman et al. (2022) presents Domain Conditional Pointwise Mutual Information to reweigh answer scores. Min et al. (2022) explores computing the probability of the input conditioned on the label. Han et al. (2023) proposes to calibrate decision boundaries. However, these studies mainly demonstrate their effectiveness for in-context learning using frozen pre-trained LMs, without addressing the intrinsic bias encoded in the LMs. Other research on mitigating intrinsic bias primarily targets removing social bias (Dinan et al., 2020; Huang et al., 2020; Cheng et al., 2021; Zhou et al., 2023), often employing costly data augmentation and re-training, and as a by-product, degrades language modeling abilities (Meade et al., 2022).

Efficiently calibrating intrinsic bias in pre-trained LMs for enhancing downstream zero/few-shot learning performance is an open research problem. We introduce a parameter-efficient intrinsic-bias calibration method leveraging automatically constructed null-input prompts, which significantly improves zero/few-shot learning of LMs.

Parameter-efficient fine-tuning (PEFT) for downstream tasks. It has been demonstrated that fine-tuning a very small portion of model parameters can achieve performance on par with fine-tuning the entire set of parameters. People propose integrating small, trainable adapter modules between model layers (Bapna and Firat, 2019; Houlsby et al., 2019), coupled with further optimization using low-rank adaptations (LoRA) (Hu et al., 2021). Some other research focuses on prompt tuning (Lester et al., 2021; Li and Liang, 2021; Gu et al., 2022; Guo et al., 2022) which only tunes continuous prompt embeddings for efficiently adapting pre-trained LMs to downstream tasks.

Our method provides a unique perspective of enhancing LM performance on downstream tasks through efficient intrinsic-bias calibration. We update only bias parameters of pre-trained LMs with null-input prompts in calibration. Contrary to adapters and LoRA which would need sufficient labeled data to learn new matrices, we do not introduce new matrices to pre-trained LMs, preserving LMs' few-shot learning capabilities. Moreover, our approach does not necessarily require target-domain data (whether labeled or unlabeled), enabling fully unsupervised deployment, particularly advantageous for zero-shot setting.

3 Null-Input Prompting for Intrinsic Bias Calibration

3.1 Task Formulation

Let $\mathcal{LM}$ be a pre-trained Masked LM. Verbalizer $V(\cdot)$ maps label y to vocabulary token. Prompt function $f_p(\cdot)$ modifies original input x_{in} into cloze-style prompt containing one <mask> token to be predicted. The output representation $\mathbf{h}_{\text{<mask>}}$ of the <mask> token is acquired from the last encoder layer after forwarding the prompt to the LM. Following Gao et al. (2021), the probability prediction of each class $y \in \mathcal{Y}$ is formulated as:

$$P(y \mid x_{\text{in}}, \mathcal{LM}) = P(V(y) \mid f_p(x_{\text{in}}), \mathcal{LM})$$
$$= \frac{\exp\left(index_{V(y)}(\mathbf{W}_{\text{lm\_head}} \cdot \mathbf{h}_{\text{<mask>}})\right)}{\sum_{j=1}^{|\mathcal{Y}|} \exp\left(index_{V(y_j)}(\mathbf{W}_{\text{lm\_head}} \cdot \mathbf{h}_{\text{<mask>}})\right)}, \tag{1}$$

where $\mathbf{W}_{\text{lm\_head}}$ is the pre-trained *masked language modeling head* weight matrix, and $index_{V(y)}$ selects the logits corresponding to the label words based on their index in LM token list.

One can probe intrinsic bias encoded in pre-trained LM by replacing x_{in} with null-meaning input $x_{\text{null}} \in \mathcal{X}_{\text{null}}$ (Zhao et al., 2021). $\mathcal{X}_{\text{null}}$ represents a set of x_{null} and we will elaborate their generation and selection in § 4. As shown by the blue bars in the upper part of Figure 1, while null-meaning inputs essentially provide no task-relevant prior information, the mean output probability associated with different labels $\bar{P}_{\mathcal{X}_{\text{null}}}(y \mid x_{\text{null}}, \boldsymbol{\mathcal{LM}})$ may exhibit significant difference attributed to model's intrinsic bias. Ideally, for bias-calibrated LM $\boldsymbol{\mathcal{LM}}_{calib}$, the expectation of output distribution conditioned on null-meaning inputs should be uniform across all label words, i.e.,

$$\mathbb{E}_{\mathcal{X}_{\text{null}}}[P(y \mid x_{\text{null}}, \boldsymbol{\mathcal{LM}}_{calib}; \forall y \in \mathcal{Y})] = \frac{1}{|\mathcal{Y}|}. \tag{2}$$

We aim to calibrate intrinsic bias by updating LM to minimize this distribution disparity which we quantify using differentiable KL divergence as:

$$\begin{aligned} &D_{\mathcal{KL}}\left(U(\mathcal{Y}) \,||\, \bar{P}_{\mathcal{X}_{\text{null}}}(\mathcal{Y})\right) \\ &= \sum_{y \in \mathcal{Y}} \left(1/|\mathcal{Y}| \cdot \log \frac{1/|\mathcal{Y}|}{\bar{P}_{\mathcal{X}_{\text{null}}}(y)}\right) \\ &= \log(1/|\mathcal{Y}|) - (1/|\mathcal{Y}|) \cdot \sum_{y \in \mathcal{Y}} \log \bar{P}_{\mathcal{X}_{\text{null}}}(y), \end{aligned} \tag{3}$$

where $U(\mathcal{Y})$ denotes uniform probability distribution and $\bar{P}_{\mathcal{X}_{\text{null}}}(y)$ represents the simplified form of $\bar{P}_{\mathcal{X}_{\text{null}}}(y \mid x_{\text{null}}, \boldsymbol{\mathcal{LM}})$.

3.2 Update Only Bias Parameters

While intrinsic bias may be encoded across various parts of pre-trained LMs, one question arises: is it essential to update the entire model, or is there a more efficient alternative that can achieve comparable effectiveness in intrinsic bias calibration? We propose to only update bias parameters $\boldsymbol{B_{LM}}$, with the following rationale: (i) $\boldsymbol{B_{LM}}$ constitutes less than 0.1% of total LM parameters, offering significant memory and computation cost saving compared to updating entire LM. (ii) Weight parameters $\boldsymbol{W_{LM}}$[1] may carry crucial pre-existing knowledge for language modeling, which risks impairment with a full model update (Meade et al., 2022). $\boldsymbol{B_{LM}}$, often overlooked in LM research, serves as offsets in DNN layers. Strategic updates may counteract intrinsic bias while potentially preserving language modeling abilities. (iii) Empirical research on efficient fine-tuning has demonstrated the important role of bias parameters in LMs (Ben Zaken et al., 2022; Logan IV et al., 2022).

We update $\boldsymbol{B_{LM}}$ using gradient descent to minimize the dissimilarity between output probability distribution from the LM conditioned on null-meaning inputs and uniform probability distribution $U(\mathcal{Y})$. We formulate a customized KL divergence loss $\mathcal{L}$, including both divergence of individual null-input's output distribution $P_i(\mathcal{Y})$ with respect to $U(\mathcal{Y})$, and batch-averaged distribution $\bar{P}_N(\mathcal{Y})$ with respect to $U(\mathcal{Y})$, as:

$$\begin{aligned} \mathcal{L} = &\frac{1}{N} \sum_{i=1}^{N} D_{\mathcal{KL}}\big(U(\mathcal{Y}) \,||\, P_i(\mathcal{Y})\big) \\ &+ D_{\mathcal{KL}}\big(U(\mathcal{Y}) \,||\, \bar{P}_N(\mathcal{Y})\big), \end{aligned} \tag{4}$$

where N is the batch size of null-meaning inputs. Incorporating the second term in the loss function promotes calibration stability and aligns with the objective of Equation 2.

3.3 Early Stopping of Calibration

We aim to obtain LM with improved zero/few-shot performance at the calibration stopping point. An overly calibrated model may simply produce uniform probability predictions regardless of input information. To avoid this, we develop specialized early stopping strategies depending on whether the downstream task is zero-shot or few-shot.

For *zero-shot* downstream tasks. Determining the calibration stopping point for optimal zero-shot learning performance is challenging due to the absence of labeled data for validation during calibration. To discern the patterns of a good stopping point, we first conduct empirical experiments by validating LM zero-shot performance on the entire test dataset after each calibration batch (consisting of N null-meaning inputs) across different calibration learning rates (Figure 7 in Appendix A). As shown in Figure 2, with optimal calibration learning rate, model performance exhibits significant improvements in the first one/few calibration batches with low variance, and then starts to degrade and becomes unstable. The low performance and instability at the calibration tail confirm our assumption on the detrimental effects of excessive calibration on LM's modeling abilities. Notably, calibration with only one batch of null inputs (indicated by the red vertical line in Figure 2) delivers

[1] $\boldsymbol{W_{LM}}$ also includes embedding parameters in our context.

consistent and significant improvement compared to the original LM (although might not be the best improvement). Therefore, for enhancing LM zero-shot performance, we directly adopt the *One-batch Calibration* as the early stopping criterion.

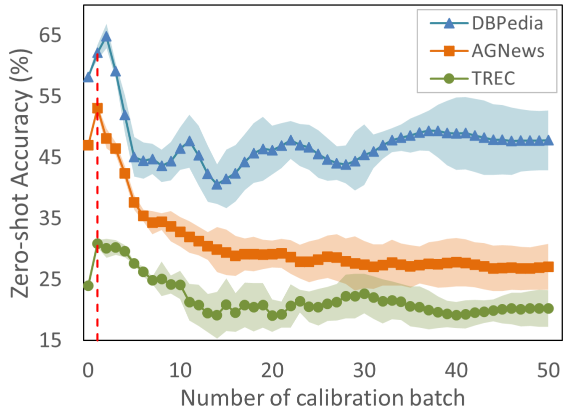

Figure 2: Empirical experiments show the impact of calibration on zero-shot learning performance as the number of calibration batches increases (batch size is 32). The intersections of the curves and red vertical line signify the outcomes of the first calibration batch.

For *few-shot* downstream tasks. With the acquisition of a few labeled downstream data, the previous challenge of lacking validation for determining the stopping point in the calibration process is alleviated. We utilize the small amount of labeled data as validation dataset $\mathcal{D}_{\text{val}}^{\text{calib}}$ to set a stopping criterion for calibration. Additionally, we take into account above-mentioned empirical findings that, for some tasks, stopping at one batch of calibration yields optimal LM performance. Relying on the limited size of $\mathcal{D}_{\text{val}}^{\text{calib}}$ might fail to identify such stopping points. To this effect, we store both $LM_{\text{calib}}^{\text{one\_batch}}$ (obtained from one-batch stopping) and $LM_{\text{calib}}^{\text{val}}$ (obtained from validation-based stopping) for downstream few-shot leaning tasks. Since $LM_{\text{calib}}^{\text{one\_batch}}$ is stored in the process of obtaining $LM_{\text{calib}}^{\text{val}}$, this will not result in additional computation overhead. Memory overhead is minimal, as it only requires storing an additional set of updated bias parameters.

We summarize our method for intrinsic bias calibration in Algorithm 1 (Appendix A).

4 Auto-Construct Null-Input Prompt

4.1 Generate Null-Meaning Input

We employ null-meaning inputs to probe the intrinsic bias of pre-trained LMs, and then use those bias-reflected outputs to calibrate the LMs. Crafting a diverse set of null-meaning inputs $\mathcal{X}_{\text{null}}$ for an averaged output helps prevent overfitting to sub-optimal instances, thereby contributing to the effectiveness of calibration. To enable cost-effective acquisition of various null-meaning data, we utilize GPT-4 API for automatic generation with instructions such as *"Please generate null meaning symbols, words, phrases, and sentences, in total <Number>."*. This process is task-agnostic, generating data that contains null information with respect to any downstream task. Note that null information is not equivalent to neutral sentiment, as it carries no inherent meaning or contextual sentiment implications. We further validate this through t-SNE (van der Maaten and Hinton, 2008) visualization in Appendix A Figure 6.

Generated null-meaning input x_{null}	$P_{nsp}(x_{\text{null}}, ans)$
This is an example sentence.	0.9996
A message without purpose.	0.9979
Words without message.	0.9809
123abc	0.0267
@#$%^&()-_=+[]{}*	0.0145
//////////////////	0.0008

Table 1: Some examples of generated null-mean inputs. In this case, *"It is <mask>."* is used as the answer format *ans*. The green/yellow numbers represent higher/lower NSP probabilities.

4.2 Select x_{null} and Build Null-Input Prompt

We construct null-input prompt $f_p(x_{\text{null}})$ by concatenating the generated null-meaning input with an answer format *ans*. For consistency, the answer format (e.g., *"It is <mask>."*) is the same as the one intended for use in the downstream task. Some examples are shown in the upper part of Figure 1.

To pursue better cohesive integration of the *"null"* information into the prompts, we prioritize the null-meaning inputs, with which the answer format exhibits higher Next Sentence Prediction (NSP) probability (Devlin et al., 2019). Specifically, after we generate a large set of null-meaning inputs $\{x_{\text{null\_1}}, x_{\text{null\_2}}, \ldots, x_{\text{null\_k}}\}$ and the answer format *ans* is selected, we employ BERT-large model (Devlin et al., 2019) to predict NSP $P_{nsp}(x_{\text{null}}, ans)$ and sort null-meaning inputs by their probabilities. Table 1 shows some generated x_{null}, with which a specific answer format presents high/low NSP scores. After the sorting, we retain the top 80% x_{null} instances (800 in total), which maintains the diversity among the selected samples. We observed that null inputs

	In-context lrn no demo[†]			In-context lrn with demo			Prompt FT no demo			Prompt FT with demo		
	NoCal	OutCal	IntrCal	NoCal	OutCal	IntrCal	NoCal	OutCal	IntrCal	NoCal	OutCal	IntrCal
AGNews	$47.0_{0.0}$	$54.3_{1.0}$	**$54.5_{0.6}$**	$79.7_{0.8}$	$78.8_{3.3}$	**$82.4_{0.9}$**	**$89.1_{0.9}$**	$86.3_{1.6}$	$89.0_{0.8}$	$86.9_{2.8}$	$87.5_{1.3}$	**$89.3_{0.9}$**
DBPedia	$58.2_{0.0}$	$54.1_{1.9}$	**$61.8_{0.6}$**	$92.6_{0.6}$	$94.0_{0.9}$	**$94.8_{0.7}$**	$98.2_{1.3}$	$99.0_{0.5}$	**$99.0_{0.1}$**	$98.6_{0.3}$	$98.5_{0.2}$	**$98.9_{0.3}$**
TREC	$24.0_{0.0}$	$29.4_{2.1}$	**$31.1_{0.5}$**	$48.3_{1.4}$	$42.5_{3.4}$	**$48.6_{2.2}$**	$85.0_{7.4}$	$82.2_{2.0}$	**$89.3_{4.5}$**	$87.6_{2.5}$	$74.2_{4.0}$	**$89.7_{1.0}$**
Subj	$50.8_{0.0}$	**$64.0_{2.7}$**	$62.7_{0.8}$	$47.2_{0.2}$	$55.0_{1.3}$	**$63.5_{2.3}$**	$91.2_{0.9}$	$88.2_{2.5}$	**$93.2_{1.2}$**	$91.4_{3.3}$	$93.0_{0.8}$	**$94.3_{0.2}$**
SST-5	$31.5_{0.0}$	$33.0_{2.1}$	**$37.5_{0.4}$**	$34.4_{1.7}$	$31.2_{2.6}$	**$36.6_{1.0}$**	$47.8_{4.6}$	$45.3_{2.8}$	**$49.9_{2.7}$**	$47.1_{1.9}$	$42.6_{4.0}$	**$50.0_{1.7}$**
Laptop	$54.6_{0.0}$	$58.3_{2.5}$	**$59.6_{1.9}$**	$50.8_{1.0}$	$65.1_{2.7}$	**$67.4_{1.7}$**	$74.3_{1.4}$	$74.3_{1.6}$	**$74.9_{2.9}$**	$76.8_{1.0}$	$75.6_{1.4}$	**$78.7_{1.4}$**
Restaurant	$68.6_{0.0}$	$72.0_{4.9}$	**$72.8_{1.6}$**	$69.8_{1.1}$	**$74.3_{1.6}$**	$74.0_{1.0}$	$79.7_{2.2}$	$79.0_{1.0}$	**$82.0_{0.9}$**	$78.4_{4.9}$	$79.0_{5.5}$	**$79.8_{4.5}$**
Twitter	$19.7_{0.0}$	$43.4_{4.1}$	**$51.7_{0.4}$**	$21.0_{0.5}$	$40.7_{5.4}$	**$49.4_{2.7}$**	$51.7_{2.9}$	$44.1_{3.9}$	**$57.0_{4.2}$**	$57.7_{2.8}$	$50.3_{4.2}$	**$59.3_{2.3}$**
Average	44.3	51.1	**54.0**	55.5	60.2	**64.6**	77.1	74.8	**79.3**	78.1	75.1	**80.0**

Table 2: Result comparisons among NoCal (LM-BFF Gao et al., 2021; no calibration), OutCal (output calibration) and IntrCal (ours; intrinsic-bias calibrated LM) using RoBERTa-large. We report the mean and standard deviation of performance in 8 classification datasets with 4 prompt-learning methods. "In-context lrn" refers to in-context learning and "Prompt FT" refers to prompt-based fine-tuning. "with/no demo" denotes incorporating/not incorporating demonstrations in prompts. In-context lrn no demo[†] is zero-shot learning, while the other three are few-shot learning.

with lower NSP scores are typically randomly-combined alphabet letters and symbols. These samples may have minimal occurrences in pre-training corpora. The low NSP scores can be attributed to RoBERTa's lack of comprehension of their meanings in context. Their representations extracted by LM might have high variance, which might impact the stability and effectiveness of calibration. We show calibration with x_{null} selection strategy further improves LM performance in § 5.2 Table 3.

5 Experiments

We conduct extensive experiments on 8 English datasets, including sentiment analysis and topic classification.[2] They consist of 5 sentence-level datasets potentially impacted by *common token bias*: AGNews (Zhang et al., 2015), DBPedia (Lehmann et al., 2015), TREC (Voorhees and Tice, 2000), Subj (Pang and Lee, 2004), SST-5 (Socher et al., 2013) and 3 aspect-level sentiment analysis datasets likely subject to *association bias*: Restaurant and Laptop reviews from SemEval 2014 Task (Pontiki et al., 2014), Twitter (Dong et al., 2014). For aspect-level datasets, the task is to predict sentiments associated with the marked aspects in each sentence. More details are in Appendix A Table 7.

5.1 Evaluation Protocol

We evaluate the effectiveness of our intrinsic-bias calibration method on enhancing Masked LMs zero/few-shot learning performance with 4 prompt learning methods: in-context learning and prompt-based fine-tuning, both with and without demonstration. We follow the prompt-based fine-tuning and demonstration method of Gao et al. (2021).

We conduct calibration with 5 different seeds, and for the few-shot setting, we randomly sample 5 different groups of training and validation sets (K samples per class). We report the mean and standard deviation of LM performance. For the 5 sentence-level classification tasks, we use *accuracy* as the metric. For the 3 aspect-level classification tasks, because of the imbalance in test set, we use *weighted* F_1 for a balanced evaluation. Details of calibration and prompt learning are in Appendix A.

We present our main results using RoBERTa-large, and $K = 16$ for few-shot setting. Results of using RoBERTa-base, $K = \{2, 4, 8\}$, and different prompt templates are in Appendix B (Table 10, Table 11 and Figure 8).

5.2 Main Results

In Table 2, we compare our results of **IntrCal** (intrinsic bias calibration) with reproduced results of: (1) **NoCal**: No calibration. Use LM-BFF (Gao et al., 2021) to compute $P(y \mid x_{\text{in}})$ for predictions. (2) **OutCal**: Output calibration. OutCal computes $\frac{P(y \mid x_{\text{in}})}{P(y \mid x_{\text{domain}})}$ instead of $P(y \mid x_{\text{in}})$ to counteract surface form competition and bias (Zhao et al., 2021; Holtzman et al., 2022). Note that OutCal was originally demonstrated for in-context learning with GPT models, while here, we apply the method in Masked LMs for fair comparisons.

In addition to NoCal and OutCal, we compare

[2]We mainly focus on single-sentence tasks, which aligns with the use of single-sentence null inputs for calibration. The alignment may enhance calibration effectiveness. We also experiment on sentence-pair tasks in Appendix B Table 15 and demonstrate better performance after calibration.

our results with those reproduced from *NoisyTune* (Wu et al., 2022), *NSP-BERT* (Sun et al., 2022) and *Perplection* (Lu et al., 2023), as detailed in Appendix B.1 (Table 8, 9). The superior performance further validates the effectiveness of our method.

In-context learning results. OutCal has significantly improved LM zero/few-shot performance compared to NoCal. Our method (IntrCal) further outperforms OutCal by a large margin: 2.9% and 8.3% absolute in zero-shot learning & 4.4% and 8.7% absolute in few-shot learning, in terms of average and best-case improvement. This demonstrates the advantages of intrinsic bias calibration over attempting to counteract bias solely at the output. Moreover, OutCal exhibits higher variance in performance due to its sensitivity to human-crafted domain-relevant strings x_{domain}. Using certain x_{domain} instances may not accurately capture the bias of LMs, resulting in under-calibration or over-calibration and leading to the high variance. In our approach, we use a large set of auto-generated and selected x_{null} as the training set for bias calibration. This mitigates the impact of sub-optimal samples and enhances calibration robustness, contributing to more stable and reliable performance.

Prompt-based fine-tuning results. This method fine-tunes all LM parameters utilizing limited labeled data by minimizing the cross-entropy loss based on Equation 1. It greatly raises LM performance compared to in-context learning and sets up a strong baseline (i.e., NoCal). OutCal fails to surpass NoCal. We speculate that OutCal's limitation lies in its exclusive focus on offsetting bias at the output and lack of interaction with the interior of LM. This appears to impede OutCal from adapting effectively to the intricate dynamics of LM after prompt-based fine-tuning, leading to some counterproductive calibrations. In contrast, IntrCal (ours) with the aim of intrinsic bias calibration achieves superior performance with absolute gains of maximum 5.3% and average 2% compared to NoCal.

The output representations of <mask> token for label word predictions are visualized by t-SNE in Figure 3. On the left, samples from the two categories are almost mixed together, indicating that the original LM tends to bias toward one class prediction. In contrast, the right visualization demonstrates improved separability after *One-batch Calibration*(§ 3.3), which explains the significant performance enhancement achieved by our intrinsic-bias calibration method.

	In-context lrn no demo		Prompt FT no demo	
	UnSel. x_{null}	*Sel.* x_{null}	*UnSel.* x_{null}	*Sel.* x_{null}
AGNews	$53.1_{0.6}$	$\mathbf{54.5}_{0.6}$	$87.8_{1.7}$	$\mathbf{89.0}_{0.8}$
DBPedia	$\mathbf{62.1}_{1.2}$	$61.8_{0.6}$	$98.7_{0.2}$	$\mathbf{99.0}_{0.1}$
TREC	$30.9_{0.6}$	$\mathbf{31.1}_{0.5}$	$88.5_{3.5}$	$\mathbf{89.3}_{4.5}$
Subj	$60.5_{3.2}$	$\mathbf{62.7}_{0.8}$	$92.8_{1.6}$	$\mathbf{93.2}_{1.2}$
SST-5	$35.5_{1.7}$	$\mathbf{37.5}_{0.4}$	$48.7_{4.2}$	$\mathbf{49.9}_{2.7}$

Table 3: Benefits from null-meaning input x_{null} selection strategy (§ 4.2). *UnSel.* signifies using all GPT-generated x_{null} in calibration, while *Sel.* denotes selecting top x_{null} based on the sorting of $P_{nsp}(x_{\text{null}}, ans)$.

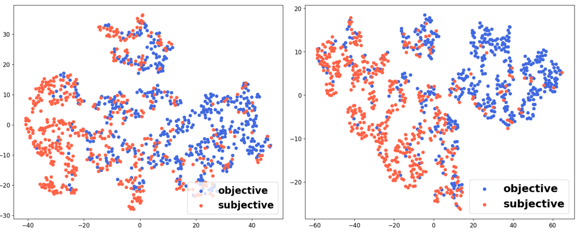

Figure 3: t-SNE visualization for output representations of <mask> token. **Left** is obtained from original LM; **Right** is obtained from the LM after *One-batch Calibration*. Two colors denote the two classes in Subj task.

	In-context lrn no demo		Prompt FT no demo	
	$\boldsymbol{W_{LM}} + \boldsymbol{B_{LM}}$	$\boldsymbol{B_{LM}}$	$\boldsymbol{W_{LM}} + \boldsymbol{B_{LM}}$	$\boldsymbol{B_{LM}}$
AGNews	$53.5_{0.8}$	$\mathbf{54.5}_{0.6}$	$\mathbf{89.3}_{0.8}$	$89.0_{0.8}$
DBPedia	$\mathbf{63.2}_{0.9}$	$61.8_{0.6}$	$99.0_{0.5}$	$\mathbf{99.0}_{0.1}$
TREC	$\mathbf{31.3}_{0.8}$	$31.1_{0.5}$	$87.6_{2.8}$	$\mathbf{89.3}_{4.5}$
Subj	$53.3_{0.6}$	$\mathbf{62.7}_{0.8}$	$\mathbf{93.7}_{0.6}$	$93.2_{1.2}$
SST-5	$33.5_{0.4}$	$\mathbf{37.5}_{0.4}$	$49.4_{0.7}$	$\mathbf{49.9}_{2.7}$
Laptop	$58.2_{0.8}$	$\mathbf{59.6}_{1.9}$	$\mathbf{78.1}_{1.3}$	$74.9_{2.9}$
Restaurant	$70.7_{1.8}$	$\mathbf{72.8}_{1.6}$	$81.3_{1.0}$	$\mathbf{82.0}_{0.9}$
Twitter	$\mathbf{51.8}_{0.7}$	$51.7_{0.4}$	$55.7_{2.3}$	$\mathbf{57.0}_{4.2}$
Average	51.9	**54.0**	79.3	79.3

Table 4: Performance comparisons between differently calibrated LMs. $\boldsymbol{W_{LM}} + \boldsymbol{B_{LM}}$ updates entire LM in calibration while $\boldsymbol{B_{LM}}$ only updates bias parameters. Additional results of In-context lrn/Prompt FT *with demo* are in Appendix B Table 14.

5.3 Update Entire LM vs. Only Bias Parameters in Calibration

In Table 4, we evaluate the impact of updating entire LM ($\boldsymbol{W_{LM}} + \boldsymbol{B_{LM}}$) during calibration on downstream task performance, as compared to only updating bias parameters ($\boldsymbol{B_{LM}}$). The optimal learning rate for updating entire LM is smaller (Appendix A Table 6). For in-context learning, the LM with only $\boldsymbol{B_{LM}}$ updates in calibration achieves better overall performance compared to the LM with entire parameter updates, most likely attributed to better preserved language modeling abilities (Appendix B Table 12). For prompt-based fine-tuning, two differently calibrated LMs demonstrate comparable performance, as the impact of entire-parameter calibration on the modeling ability is mitigated through task-specific fine-tuning. Considering the significant saving in memory and computation, we recommend only updating B_{LM} in calibration.

5.4 Analysis

How does intrinsic bias calibration impact downstream tasks? Our method calibrates the intrinsic bias associated with a set of task-specific label words. In this section, we explore the impact of updating LM for task-specific bias calibration on other downstream task performance. Specifically, we take the LM calibrated for one task and evaluate its performance on the other tasks as shown in Figure 4. In general, intrinsic bias calibration for one task has a minimal adverse effect on other tasks' performance (no more than 2% degradation) because of the light model updates, while remarkably enhancing LM performance on that specific task. Notably, there is consistent performance increase at bottom right, as these tasks are all sentiment classification sharing or including same label words.[3]

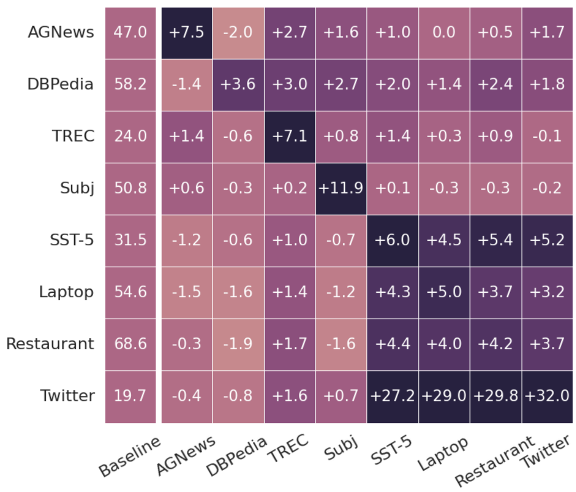

Figure 4: Impact of calibration on downstream tasks shown through the changes with respect to baseline on each column. Each row shows the zero-shot performance of one task employing: *original LM* (first column; baseline), *task-specific calibrated LM* (diagonal), *other-task calibrated LM* (other places).

[3]For aspect-level datasets, better improvement is on the diagonals (task-specific calibration), indicating our method mitigates the impact of association bias (Appendix A).

How does intrinsic bias calibration impact language modeling abilities? We employ pseudo-perplexity (Salazar et al., 2020) to evaluate language modeling for Masked LM. Following each task-specific intrinsic bias calibration, we measure pseudo-perplexity and compare the results with original RoBERTa on WikiText-2, WikiText-103 (Merity et al., 2017), and LAMBADA dataset (Paperno et al., 2016). As shown in Table 5, language modeling abilities are largely preserved after calibration due to the minimal updates to the model.

	WT-2	WT-103	LAMBADA
Original RoBERTa	6.189	7.008	24.52
+ CALIBRATION			
for_AGNews	↑0.017 6.206	↑0.029 7.037	↑0.02 24.54
for_DBPedia	↑0.008 6.197	↑0.002 7.010	↓0.22 24.30
for_TREC	↓0.027 6.162	↓0.042 6.966	↓0.27 24.25
for_Subj	↓0.021 6.168	↓0.030 6.978	↑0.08 24.60
for_SST-5	↓0.031 6.158	↓0.039 6.969	↓0.18 24.34
for_Laptop	↑0.011 6.200	↑0.002 7.010	↓0.01 24.51
for_Restaurant	↑0.055 6.244	↑0.074 7.082	↑0.13 24.65
for_Twitter	↓0.029 6.160	↓0.037 6.971	↑0.05 24.57

Table 5: Pseudo-perplexities of *original RoBERTa* and *task-specific calibrated RoBERTa* on WikiText-2 (WT-2), WikiText-103 (WT-103) and LAMBADA. We use 2000 test samples of each dataset. An increase in values (highlighted in red) indicates a reduction in language modeling abilities after calibration.

6 Conclusion

In this work, we propose a null-input prompting method to calibrate the intrinsic bias of pre-trained Masked LMs, aiming to enhance zero/few-shot learning performance in classification tasks. Our method incorporates two key features for efficiency: (1) auto-construction of null-input prompts for bias probing, leveraging a diverse set of selected null-meaning inputs easily crafted from generative Large LM; (2) updating only bias parameters for bias calibration. Experimental results show that bias-calibrated LMs demonstrate significant performance improvement for both in-context learning and prompt-based fine-tuning, with average gains of 9% and 2%, respectively. Moreover, our method outperforms output-calibration approaches, highlighting the advantage of intrinsic bias calibration. We believe this work presents a new perspective of making LMs better zero/few-shot learners via intrinsic bias calibration. Additionally, the demonstrated significance of bias parameters could provide insights for future bias-related research.

7 Limitations

While our method has achieved substantial improvement in prompt-based zero/few-shot learning, it comes with limitations that could open avenues for future research.

First, calibration is fully unsupervised in the scenario where no labeled data is available (zero-shot downstream tasks in § 3.3). Based on empirical experimental results, we adopt the conservative *One-batch Calibration* strategy to ensure a safe and consistent performance enhancement. In the future, we aim to explore more rigorous approaches to determine optimal stopping points in this scenario.

Second, we utilize RoBERTa (encoder) models for classification tasks, as encoder models may more effectively encode task-specific patterns for discriminative tasks compared to some generative LMs (Gao et al., 2021; Li et al., 2023b), as shown in Table 16. However, the relatively small size of those Masked LMs (355M parameters for RoBERTa-large) could be the ultimate limitation to their capabilities. Given the proliferation of large-scale generative (decoder) LMs and their accomplishments in tackling more challenging tasks (Thoppilan et al., 2022; Chowdhery et al., 2023; Touvron et al., 2023), we anticipate extending our method to large decoder models and validating the applicability of our findings. Furthermore, we expect to expand the scope of tasks to include regression problems (e.g., sentiment score prediction) leveraging KL divergence to measure disparities in continuous probability distributions, aiming to address bias-related challenges across diverse scenarios.

8 Ethics Statement and Broader Impact

Our work is conformant to the Code of Ethics. We appropriately cite relevant methods, models, and datasets that we use. We affirm that all datasets in our experiments are public, and no private or sensitive information is incorporated in our research. Our use of datasets and pre-trained models is consistent with their intended use. For broader impacts, our method, extending beyond calibrating common token bias and association bias, might inspire prospective research in mitigating social bias and improving the fairness of pre-trained LMs.

References

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiahao Cao, Rui Liu, Huailiang Peng, Lei Jiang, and Xu Bai. 2022. Aspect is not you need: No-aspect differential sentiment framework for aspect-based sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1599–1609.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. Fairfil: Contrastive neural debiasing method for pretrained text encoders. *arXiv preprint arXiv:2103.06413*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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

Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020.

Queens are powerful too: Mitigating gender bias in dialogue generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8173–8188, Online. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics.

Xu Guo, Boyang Li, and Han Yu. 2022. Improving the sample efficiency of prompt tuning with domain adaptation. *arXiv preprint arXiv:2210.02952*.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2023. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2022. Surface form competition: Why the highest probability answer isn't always right. *arXiv preprint arXiv:2104.08315*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. 2020. Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 65–83, Online. Association for Computational Linguistics.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2022. Contrastive learning for prompt-based few-shot language learners. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5577–5587, Seattle, United States. Association for Computational Linguistics.

Masahiro Kaneko and Danushka Bollegala. 2021. Debiasing pre-trained contextualised embeddings. *arXiv preprint arXiv:2101.09523*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yingji Li, Mengnan Du, Rui Song, Xin Wang, and Ying Wang. 2023a. A survey on fairness in large language models. *arXiv preprint arXiv:2308.10149*.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu-lee Wang, Qing Li, and Xiaoqin Zhong. 2023b. Label supervised llama finetuning. *arXiv preprint arXiv:2310.01208*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

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

Robert Logan IV, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2022.

Cutting down on prompts and parameters: Simple few-shot learning with language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2824–2835, Dublin, Ireland. Association for Computational Linguistics.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Jinghui Lu, Dongsheng Zhu, Weidong Han, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023. What makes pre-trained language models better zero-shot learners? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2288–2303, Toronto, Canada. Association for Computational Linguistics.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1878–1898, Dublin, Ireland. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *arXiv preprint cs/0409058*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

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

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

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

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Irene Solaiman and Christy Dennison. 2021. Process for adapting language models to society (palms) with values-targeted datasets. *Advances in Neural Information Processing Systems*, 34:5861–5873.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. NSP-BERT: A prompt-based few-shot learner through an original pre-training task —— next sentence prediction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3233–3250, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. 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*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

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

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2022. NoisyTune: A little noise can help you finetune pretrained language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 680–685, Dublin, Ireland. Association for Computational Linguistics.

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

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Fan Zhou, Yuzhou Mao, Liu Yu, Yi Yang, and Ting Zhong. 2023. Causal-debias: Unifying debiasing in pretrained language models and fine-tuning via causal invariant learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4227–4241.

A Experimental Details

Prompts with or without demonstrations. Table 7 shows the prompt templates and label words of each dataset we use for main experiments.

For downstream tasks, in few-shot setting, task-specific example-label pairs (i.e., demonstrations) can be incorporated in the context to enhance the LM's comprehension. While in zero-shot setting, no labeled data is available and thereby no demonstrations.

For calibration, demonstrations are either absent from or added to null-input prompts, consistent with their exclusion from or inclusion in prompts for downstream tasks. An example of a null-input prompt without demonstration is:

<s> An empty sentence. <u>It is <mask>.</u> </s>

<s> and *</s>* respectively denote `<cls>` token and `<sep>` token in RoBERTa. In the other case, we incorporate demonstrations retrieved from the small training set into the null-input prompt such as:

<s> An empty sentence. <u>It is <mask>.</u> </s>
Compellingly watchable. <u>It is great.</u> </s>
The film is strictly routine. <u>It is terrible.</u> </s>

Association-bias calibration for aspect-level task. For aspect-level sentiment analysis, e.g., *"Wonderful food but poor <u>service</u>. <u>Service</u> was <mask>."*, the answer contains the aspect word *"service"*. Because the model makes sentiment predictions for specific aspect words, the task is likely subject to *association bias* (§ 2). For association-bias calibration, the only difference is that we incorporate various aspect words in the answer format (e.g., *"<aspect words> was <mask>."*) when constructing null-input prompts. One can either leverage GPT-4 to generate in-domain aspect words (e.g., for restaurant reviews, the generated aspect words could be *menu, food*, etc.), or simply employ the aspect words in the original training dataset. In this work, we choose the latter option. Due to the variability of <*aspect words*> in the answer format, sorting null-meaning inputs by NSP score can yield different results. To this effect, we do not apply x_{null} selection strategy (§ 4.2) for aspect-level task, and instead keep all the generated x_{null}.

Null-meaning inputs for *One-batch Calibration*. For zero-shot downstream tasks, since only one batch of null-meaning inputs is required for calibration in our early-stopping criterion (§ 3.3), we select the $Top\text{-}N\{P_{nsp}(x_{\text{null}}, ans)\}$ x_{null} from $\mathcal{X}_{\text{null}}$, where N is batch size. We prioritize these samples as our observations show that null-meaning inputs with higher $P_{nsp}(x_{\text{null}}, ans)$ exhibit higher attention scores between the null input and `<mask>`, as demonstrated in Figure 5. This indicates more effective conveyance of the *"null"* information to the placeholder `<mask>`, which could facilitate LM deciphering the *"null"* patterns of the prompts and benefit calibration.

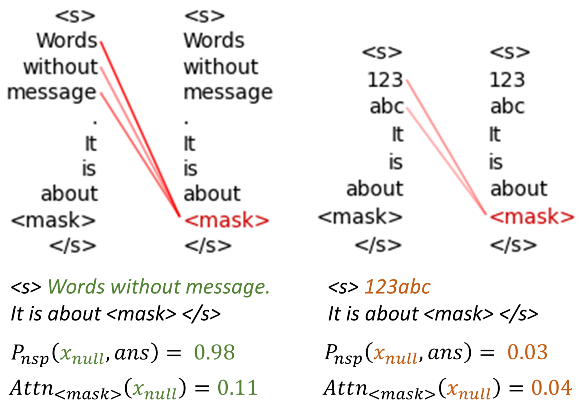

Figure 5: Visualization of attention score by the depth of color in the connecting lines. We only show the attention between `<mask>` token and null-meaning input x_{null}. $Attn_{<mask>}(x_{\text{null}})$ is the attention score of `<mask>` on x_{null}, averaged over encoder layers and attention heads. **Left**: Higher attention score indicates enhanced pattern extraction from x_{null} which has higher $P_{nsp}(x_{\text{null}}, ans)$.

Hyper-parameters. In calibration stage, we shuffle the null-input prompts and conduct gradient descent on $\boldsymbol{B_{LM}}$ (or $\boldsymbol{W_{LM}}$ + $\boldsymbol{B_{LM}}$ as comparative experiment) with 5 different seeds to account for calibration variance. There are two main hyper-parameters for calibration: (1) x_{null} batch size N; (2) calibration learning rate lr_{calib}. We conduct grid search on $N = \{8, 16, 32\}$ and lr_{calib} from $1e-6$ to $1e-3$, and obtain the best settings: $N = 32$ and lr_{calib} as shown in Table 6.

Calibrated LMs are applied in downstream tasks with prompt-learning methods. We use the same hyper-parameters as Gao et al. (2021) for prompt learning. We evaluate on each task's original test set, except for AGNews and DBPedia, where we randomly sample 2000 test examples.

We use PyTorch (Paszke et al., 2019) and public HuggingFace Transformers library (Wolf et al., 2020), and conduct all the experiments with one NVIDIA V100 GPU in Google Colab.

	Calibration (lr_{calib})		Prompt FT (downstream)
	W_{LM} + B_{LM}	B_{LM}	
No demo	$1e-5$	$1e-3$	$1e-5$
With demo	$1e-6$	$1e-4$	$1e-5$

Table 6: Optimal learning rates for calibration and downstream prompt-based fine-tuning (Prompt FT). With/No demo denotes adding/not adding demonstrations in prompts.

Algorithm 1 Null-input prompting for calibration

Inputs:
Downstream task: *zero_shot* or *few_shot*
Null-input prompts: $\{N_{\text{prompt}}\}$
(Val. data in Calibration: $\mathcal{D}^{\text{calib}}_{\text{val}} \leftarrow \mathcal{D}^{\text{downstrm}}_{\text{train}}$)
▷ Only when downstream task is *few_shot*.
▷ Downstream training dataset $\mathcal{D}^{\text{downstrm}}_{\text{train}}$ constitutes K samples per class.
Output:
$LM^{\text{one\_batch}}_{\text{calib}}$ for *zero_shot*
$LM^{\text{one\_batch}}_{\text{calib}}$ & $LM^{\text{val}}_{\text{calib}}$ for *few_shot*

```
for batch in {N_prompt} do
    P = LM(batch)                         ▷ Null input prompting
    L = D_KL(U || P)                      ▷ Unif. distribution U
    B_LM ← B_LM − α · ∂L/∂B_LM            ▷ Freeze W_LM
    if first batch then
        Save LM_calib^one_batch
    end if
    if downstream is zero_shot then break
    end if
    if better Compute_Metric(D_val^calib) then
        Save LM_calib^val
    end if
end for
```

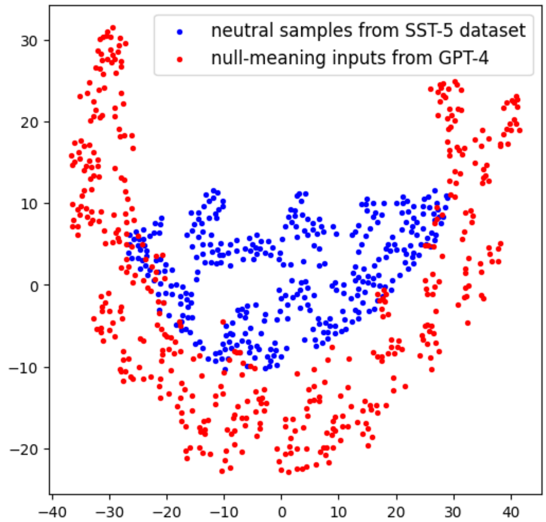

Figure 6: t-SNE visualization of output representations for null-meaning inputs generated from GPT-4 (red) compared to neutral samples from SST-5 dataset (blue). We utilize the pre-trained sentiment analysis model (Loureiro et al., 2022) to obtain the embeddings. The different distributions validate that null information is not equivalent to neutral sentiment.

Dataset	Task Type	Prompt Template	Label Words
AGNews	News topic classification	{Sentence} It is about <mask>.	World / Sports / Business / Technology
DBPedia†	Ontology classification	{Sentence} It is about <mask>.	Company / Artist / Building / Nature
TREC	Question classification	{Sentence} It is about <mask>.	Number / Location / Person / Description / Entity / Expression
Subj	Subjectivity classification	{Sentence} This is <mask>.	objective / subjective
SST-5	Movie sentiment analysis	{Sentence} The movie was <mask>.	terrible / bad / okay / good / great
Laptop	Aspect level sentiment analysis	{Sentence} {Aspect words} was <mask>.	terrible / okay / great
Restaurant	Aspect level sentiment analysis	{Sentence} {Aspect words} was <mask>.	terrible / okay / great
Twitter	Aspect level sentiment analysis	{Sentence} {Aspect words} was <mask>.	terrible / okay / great

Table 7: Prompt templates and label words of the eight datasets in our experiments for main results. For DBPedia†, we use four classes out of the total fourteen classes.

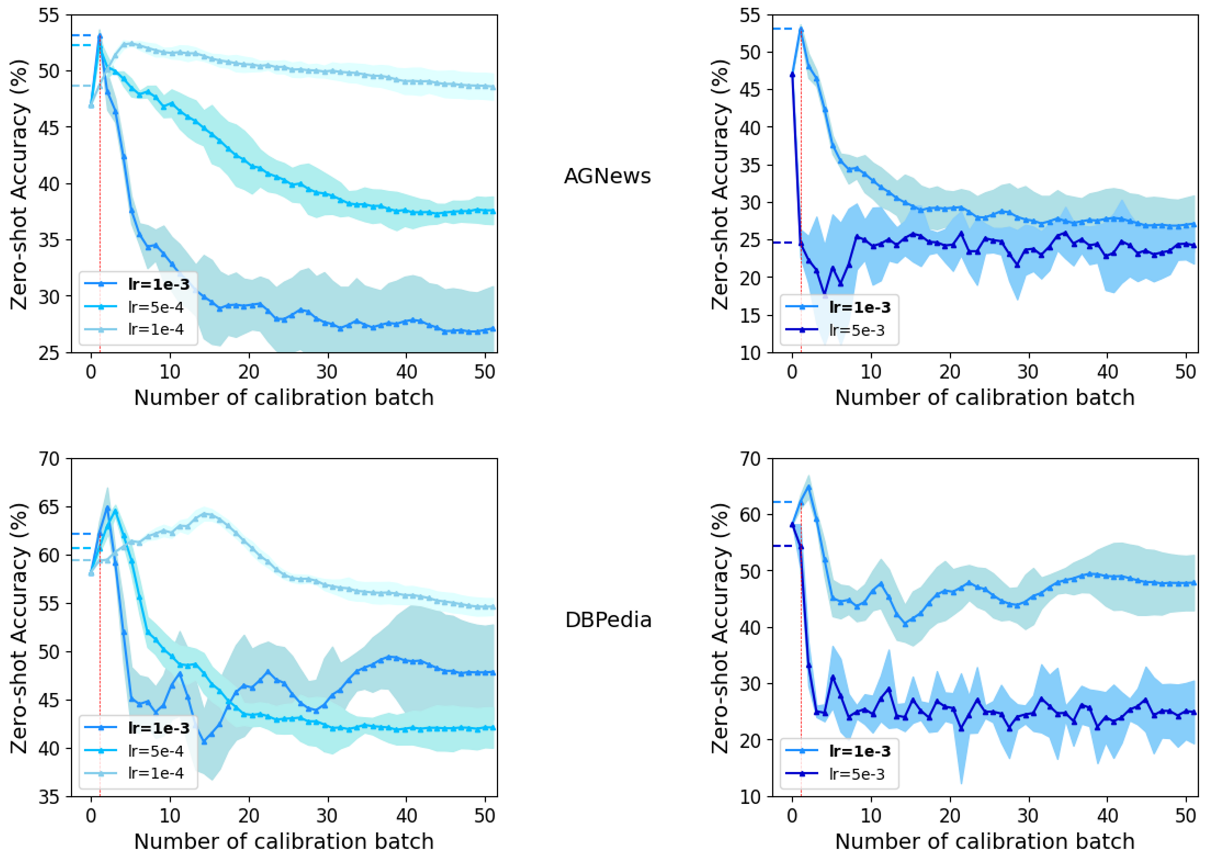

Figure 7: Empirical experiments show the impact of calibration on zero-shot learning performance across *different calibration learning rates* lr_{calib}, with a fixed batch size of 32. Only $\boldsymbol{B_{LM}}$ is updated in calibration. We identify the optimal $lr_{calib} = 1e-3$ across all datasets and illustrate with AGNews dataset (top two figures) and DBPedia dataset (bottom two figures). A smaller learning rate (left figures) consistently yields less performance improvement, considering both peak accuracy and accuracy after the first calibration batch (the intersections of the curves and red vertical line). A larger learning rate (right figures) consistently degrades performance.

B Additional Results

B.1 Performance Comparison with NSP-BERT, Perplection and NoisyTune

We additionally choose NSP-BERT (Sun et al., 2022) and Perplection (Lu et al., 2023) as *in-context learning* comparison baselines and NoisyTune (Wu et al., 2022) as *prompt-base fine-tuning* comparison baseline. NSP-BERT constructs potential answers using each label word and predict Next Sentence Prediction (NSP) probability between the input and each answer. Perplection proposes perplexity-based selection method for zero-shot prompt learning. NoisyTune demonstrates that adding noise to pre-trained LMs benefits fine-tuning on downstream tasks. We re-implement their methods with the same settings as ours for fair comparisons. As shown in Table 8 and Table 9, our method achieves superior results in almost all datasets.

Furthermore, our method consistently outperforms NoisyTune, demonstrating that the gains in prompt-based fine-tuning with our method are not solely a result of perturbing LM parameters. This confirms the efficacy of intrinsic bias calibration in enhancing LM performance.

	Zero-shot in-context learning		
	NSP-BERT	Perplection	IntrCal
AGNews	52.4	49.3	**54.5**
DBPedia	58.4	59.6	**61.8**
TREC	**32.4**	30.8	31.1
Subj	60.3	59.9	**62.7**
SST-5	30.2	31.0	**37.5**
Laptop	57.3	58.2	**59.6**
Restaurant	50.4	66.5	**72.8**
Twitter	35.3	31.5	**51.7**
Average	47.1	48.4	**54.0**

Table 8: Comparison of NSP-BERT (Sun et al., 2022), Perplection (Lu et al., 2023) and IntrCal (ours) in zero-shot in-context learning.

B.2 Other Experiments

We briefly summarize the contents of each table and figure below that presents other additional results.

Table 10 contains results for performance using RoBERTa-base model.

Table 11 contains results for performance of $K = \{2, 4, 8\}$ few-shot learning.

	Prompt FT no demo		**Prompt FT with demo**	
	NoisyTune	IntrCal	NoisyTune	IntrCal
AGNews	$89.0_{1.8}$	$\mathbf{89.0}_{0.8}$	$88.4_{1.5}$	$\mathbf{89.3}_{0.9}$
DBPedia	$98.0_{0.8}$	$\mathbf{99.0}_{0.1}$	$98.6_{0.9}$	$\mathbf{98.9}_{0.3}$
TREC	$86.2_{4.3}$	$\mathbf{89.3}_{4.5}$	$87.2_{4.6}$	$\mathbf{89.7}_{1.0}$
Subj	$93.0_{1.2}$	$\mathbf{93.2}_{1.2}$	$92.9_{1.2}$	$\mathbf{94.3}_{0.2}$
SST-5	$49.4_{1.1}$	$\mathbf{49.9}_{2.7}$	$47.5_{3.5}$	$\mathbf{50.0}_{1.7}$
Laptop	$73.8_{3.2}$	$\mathbf{74.9}_{2.9}$	$75.5_{3.2}$	$\mathbf{78.7}_{1.4}$
Restaurant	$79.9_{2.7}$	$\mathbf{82.0}_{0.9}$	$78.3_{2.6}$	$\mathbf{79.8}_{4.5}$
Twitter	$51.8_{5.8}$	$\mathbf{57.0}_{4.2}$	$59.0_{1.9}$	$\mathbf{59.3}_{2.3}$
Average	77.6	**79.3**	78.4	**80.0**

Table 9: Comparison between NoisyTune (Wu et al., 2022) and IntrCal (ours) in prompt-based fine-tuning.

Figure 8 contains results for performance using different prompt templates (Table 13).

Table 12 contains results for pseudo-perplexity comparisons between updating entire LM and only updating bias parameters in calibration.

Table 14 contains results for performance comparisons between updating entire LM and only updating bias parameters in calibration.

Table 15 contains results for performance of sentence-pair datasets.

Table 16 contains results for performance comparisons between Llama-2 and RoBERTa.

Table 17 contains results for variance of probability distribution across labels before and after calibration.

	In-context lrn no demo			In-context lrn with demo			Prompt FT no demo			Prompt FT with demo		
	NoCal	OutCal	IntrCal	NoCal	OutCal	IntrCal	NoCal	OutCal	IntrCal	NoCal	OutCal	IntrCal
AGNews	37.8$_{0.0}$	36.2$_{4.6}$	**49.0**$_{0.9}$	68.4$_{0.4}$	69.7$_{4.3}$	**73.7**$_{0.3}$	88.2$_{0.3}$	87.8$_{0.6}$	**88.9**$_{1.0}$	86.7$_{0.1}$	74.2$_{4.1}$	**87.2**$_{0.1}$
DBPedia	**57.2**$_{0.0}$	50.5$_{7.1}$	54.9$_{0.1}$	56.5$_{3.4}$	78.7$_{4.4}$	**83.9**$_{0.4}$	95.2$_{2.1}$	93.5$_{5.0}$	**99.0**$_{0.4}$	97.8$_{0.9}$	96.7$_{0.8}$	**98.6**$_{0.1}$
TREC	28.2$_{0.0}$	25.4$_{4.4}$	**30.2**$_{0.1}$	41.2$_{0.3}$	39.9$_{3.8}$	**42.5**$_{1.0}$	82.5$_{10.9}$	70.3$_{2.3}$	**86.4**$_{6.5}$	85.7$_{1.8}$	80.6$_{5.0}$	**91.2**$_{0.6}$
Subj	53.6$_{0.0}$	63.6$_{1.9}$	**66.4**$_{1.8}$	50.8$_{0.2}$	67.0$_{1.7}$	**69.6**$_{0.4}$	**92.5**$_{1.3}$	91.1$_{0.4}$	91.9$_{1.7}$	90.4$_{2.1}$	92.0$_{0.2}$	**92.3**$_{0.1}$
SST-5	31.9$_{0.0}$	30.8$_{3.4}$	**32.2**$_{0.2}$	25.3$_{4.3}$	28.6$_{3.4}$	**29.8**$_{1.7}$	45.9$_{3.3}$	42.9$_{2.3}$	**48.1**$_{1.8}$	44.3$_{5.2}$	40.7$_{2.5}$	**45.8**$_{2.6}$
Laptop	56.1$_{0.0}$	56.7$_{3.8}$	**60.0**$_{0.1}$	49.2$_{0.9}$	61.5$_{2.8}$	**64.0**$_{0.6}$	75.8$_{3.4}$	73.0$_{1.3}$	**76.3**$_{1.8}$	74.8$_{0.1}$	76.0$_{0.6}$	**76.3**$_{0.5}$
Restaurant	69.8$_{0.0}$	**72.0**$_{2.9}$	69.5$_{0.5}$	67.6$_{0.7}$	70.5$_{2.4}$	**73.2**$_{0.7}$	75.5$_{6.6}$	**77.3**$_{3.4}$	77.2$_{1.1}$	74.8$_{3.3}$	75.2$_{0.7}$	**76.1**$_{3.9}$
Twitter	22.0$_{0.0}$	48.6$_{5.1}$	**52.3**$_{0.6}$	17.6$_{0.4}$	41.8$_{5.4}$	**48.4**$_{0.5}$	54.5$_{1.1}$	47.7$_{3.8}$	**57.9**$_{1.3}$	50.6$_{4.6}$	51.8$_{2.1}$	**56.0**$_{4.9}$
Average	44.6	48.0	**51.8**	47.1	57.2	**60.6**	76.3	73.0	**78.2**	75.6	73.4	**77.9**

Table 10: Result comparisons among NoCal (LM-BFF Gao et al., 2021; no calibration), OutCal (output calibration) and IntrCal (ours; intrinsic-bias calibrated LM) using RoBERTa-base. We report the mean and standard deviation of performance in 8 classification datasets with 4 prompt-learning methods.

		In-context lrn with demo		Prompt FT no demo		Prompt FT with demo	
		NoCal	IntrCal	NoCal	IntrCal	NoCal	IntrCal
2-shot	AGNews	70.4$_{6.7}$	**76.3**$_{3.6}$	76.4$_{5.4}$	**80.2**$_{8.0}$	78.2$_{1.3}$	**83.2**$_{1.1}$
	DBPedia	92.9$_{0.9}$	**94.0**$_{1.0}$	97.0$_{1.6}$	**98.4**$_{0.9}$	97.4$_{1.0}$	**97.8**$_{1.1}$
	TREC	49.8$_{4.2}$	**50.5**$_{4.0}$	49.1$_{22.6}$	**60.3**$_{9.6}$	65.2$_{9.3}$	**66.1**$_{9.3}$
	Subj	49.4$_{1.1}$	**56.2**$_{3.9}$	66.4$_{5.4}$	**82.2**$_{5.9}$	72.3$_{13.9}$	**81.5**$_{13.2}$
4-shot	AGNews	75.7$_{3.9}$	**80.3**$_{1.7}$	85.4$_{2.7}$	**87.3**$_{1.3}$	76.7$_{13.1}$	**85.9**$_{1.9}$
	DBPedia	93.0$_{0.4}$	**93.9**$_{0.4}$	97.2$_{0.8}$	**97.9**$_{1.1}$	96.4$_{1.5}$	**98.6**$_{0.6}$
	TREC	51.9$_{2.6}$	**53.2**$_{2.5}$	64.5$_{7.1}$	**67.6**$_{6.7}$	73.6$_{8.5}$	**78.2**$_{9.7}$
	Subj	48.8$_{2.2}$	**59.4**$_{3.1}$	81.4$_{3.9}$	**88.5**$_{3.2}$	78.9$_{9.3}$	**83.6**$_{7.8}$
8-shot	AGNews	79.6$_{1.0}$	**82.4**$_{1.6}$	86.9$_{1.9}$	**88.1**$_{0.4}$	85.5$_{1.7}$	**88.0**$_{1.4}$
	DBPedia	92.9$_{0.8}$	**94.2**$_{0.2}$	97.3$_{1.2}$	**98.8**$_{0.5}$	98.2$_{0.8}$	**98.6**$_{0.2}$
	TREC	47.9$_{2.2}$	**48.7**$_{2.0}$	71.6$_{4.9}$	**72.2**$_{5.1}$	75.4$_{6.2}$	**81.7**$_{5.6}$
	Subj	48.4$_{1.0}$	**60.5**$_{4.8}$	91.9$_{1.3}$	**92.7**$_{0.8}$	88.9$_{5.3}$	**92.1**$_{2.2}$

Table 11: Few-shot learning with different number of training samples ($K = \{2, 4, 8\}$) using RoBERTa-large. IntrCal (ours; intrinsic-bias calibrated LM) consistently outperforms NoCal (no calibration).

Model	**Datasets**					
	WikiText-2		**WikiText-103**		**LAMBADA**	
Original RoBERTa	6.189		7.008		24.52	
+ CALIBRATION	$W_{LM} + B_{LM}$	B_{LM}	$W_{LM} + B_{LM}$	B_{LM}	$W_{LM} + B_{LM}$	B_{LM}
for_AGNews	↑0.105 6.294	↑0.017 6.206	↑0.059 7.067	↑0.029 7.037	↑0.58 25.10	↑0.02 24.54
for_DBPedia	↑0.101 6.290	↑0.008 6.197	↑0.092 7.100	↑0.002 7.010	↑0.76 25.28	↓0.22 24.30
for_TREC	↑0.049 6.238	↓0.027 6.162	↑0.040 7.048	↓0.042 6.966	↑0.57 25.09	↓0.27 24.25
for_Subj	↑0.081 6.270	↓0.021 6.168	↑0.116 7.124	↓0.030 6.978	↑0.70 25.22	↑0.08 24.60
for_SST-5	↓0.018 6.171	↓0.031 6.158	↑0.143 7.151	↓0.039 6.969	↑0.65 25.17	↓0.18 24.34
for_Laptop	↑0.133 6.322	↑0.011 6.200	↑0.075 7.083	↑0.002 7.010	↑0.56 25.08	↓0.01 24.51
for_Restaurant	↑0.102 6.291	↑0.055 6.244	↑0.071 7.079	↑0.074 7.082	↑0.64 25.16	↑0.13 24.65
for_Twitter	↑0.204 6.393	↓0.029 6.160	↑0.096 7.104	↓0.037 6.971	↑0.39 24.91	↑0.05 24.57

Table 12: Pseudo-perplexities of original RoBERTa and task-specific calibrated RoBERTa on WikiText-2, WikiText-103 and LAMBADA. We use 2000 test samples of each dataset. An increase in values (highlighted in red) indicates a reduction in language modeling abilities after calibration. $W_{LM} + B_{LM}$ updates entire LM in calibration while B_{LM} only updates bias parameters.

Task	Prompt Templates
AGNews	{Sentence} It is about <mask>.
	{Sentence} This is about <mask>.
	{Sentence} This is on <mask>.
	{Sentence} It pertains to <mask>.
	{Sentence} In relation to <mask>.
TREC	{Sentence} It is about <mask>.
	{Sentence} Concerning <mask>.
	{Sentence} This is about <mask>.
	{Sentence} In relation to <mask>.
	{Sentence} This is on <mask>.

Table 13: The five different prompt templates used in Figure 8.

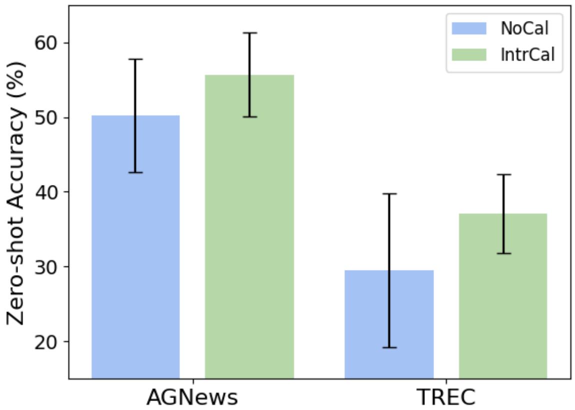

Figure 8: Performance comparison averaged on using five different prompt templates with RoBERTa-large. IntrCal (ours; intrinsic-bias calibrated LM) demonstrates significantly improved accuracy with lower variance compared to NoCal (no calibration).

	In-context lrn with demo		Prompt FT with demo	
	$W_{LM} + B_{LM}$	B_{LM}	$W_{LM} + B_{LM}$	B_{LM}
AGNews	$82.0_{0.8}$	$\mathbf{82.4}_{0.9}$	$\mathbf{89.3}_{0.6}$	$89.3_{0.9}$
DBPedia	$\mathbf{95.1}_{0.7}$	$94.8_{0.7}$	$\mathbf{99.0}_{0.1}$	$98.9_{0.3}$
TREC	$\mathbf{49.1}_{2.6}$	$48.6_{2.2}$	$88.9_{2.3}$	$\mathbf{89.7}_{1.0}$
Subj	$\mathbf{65.6}_{0.4}$	$63.5_{2.3}$	$93.9_{1.6}$	$\mathbf{94.3}_{0.2}$
SST-5	$\mathbf{37.1}_{1.0}$	$36.6_{1.0}$	$\mathbf{51.3}_{1.7}$	$50.0_{1.7}$
Laptop	$65.8_{0.3}$	$\mathbf{67.4}_{1.7}$	$77.7_{0.8}$	$\mathbf{78.7}_{1.4}$
Restaurant	$72.7_{1.2}$	$\mathbf{74.0}_{1.0}$	$\mathbf{81.4}_{3.4}$	$79.8_{4.5}$
Twitter	$45.8_{2.7}$	$\mathbf{49.4}_{2.7}$	$\mathbf{60.4}_{1.7}$	$59.3_{2.3}$
Average	64.2	**64.6**	**80.2**	80.0

Table 14: Performance comparisons between differently calibrated LMs using RoBERTa-large. $W_{LM} + B_{LM}$ updates entire LM in calibration while B_{LM} only updates bias parameters. This table (prompt learning *with* demonstrations) is the supplement to § 5.3 Table 4 (prompt learning *without* demonstrations).

	In-context lrn no demo		Prompt FT no demo	
	NoCal	IntrCal	NoCal	IntrCal
MNLI	$32.7_{0.0}$	$\mathbf{37.7}_{0.7}$	$67.9_{2.1}$	$\mathbf{68.6}_{1.9}$
SNLI	$33.6_{0.0}$	$\mathbf{36.7}_{0.9}$	$77.4_{2.8}$	$\mathbf{78.5}_{2.3}$
MRPC	$51.1_{0.0}$	$\mathbf{53.6}_{0.2}$	$73.6_{4.3}$	$\mathbf{74.9}_{1.4}$
QQP	$50.8_{0.0}$	$\mathbf{54.6}_{0.2}$	$65.2_{3.5}$	$\mathbf{66.2}_{3.3}$

Table 15: Benchmark on sentence-pair datasets, MNLI (Williams et al., 2018), SNLI (Bowman et al., 2015), MRPC (Dolan and Brockett, 2005), QQP (Wang et al., 2018). NoCal denotes no-calibration (baseline) and IntrCal denotes our method. Our method demonstrates effectiveness on sentence-pair datasets. The overall low performance of in-context learning can be attributed to two main factors: (1) RoBERTa's inherent limited capabilities when using in-context learning for the more difficult tasks, which is significantly improved with prompt-based fine-tuning. (2) The misalignment between these sentence-pair datasets and the use of single-sentence null inputs for calibration, which could impact the effectiveness of calibration.

	Llama-2 (7B)	RoBERTa-large (355M)
AGNews	44.1	**47.0**
DBPedia	47.1	**58.2**
TREC	**42.0**	24.0
Subj	49.8	**50.8**
SST-5	29.3	**31.5**
Laptop	48.5	**54.6**
Restaurant	65.4	**68.6**
Twitter	**25.5**	19.7

Table 16: Comparison between Llama-2 (7B parameters) (Touvron et al., 2023) and RoBERTa-large (355M parameters) on zero-shot in-context learning performance for classification tasks. Llama-2 does not consistently outperform RoBERTa in these tasks.

	AGNews	DBPedia	TREC	Subj	SST-5
Orig. LM	0.033	0.130	0.025	0.195	0.011
Calib. LM	0.022	0.025	0.011	0.112	0.011

Table 17: We calculate the **variance** of probability distribution across labels conditioned on null-meaning inputs, i.e., $Var\left(\bar{P}_{\mathcal{X}_{\text{null}}}(\mathcal{Y})\right)$, before and after calibration. A smaller variance indicates that a distribution is closer to uniform distribution. Orig. LM denotes original LM, and Calib. LM denotes the LM after *One-batch Calibration* (§ 3.3). The decreasing variance in each task after calibration demonstrates that our method promotes the establishment of equitable LMs.