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 prompt pre-trained LMs for intrinsic bias probing. 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., 2023). To improve efficiency in both data generation and model updates, we propose leveraging auto-generated *null-meaning inputs* to prompt LMs for intrinsic bias probing, and subsequently updating only *bias parameters* $\boldsymbol{B_{LM}}$ of LMs for bias calibration. 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,

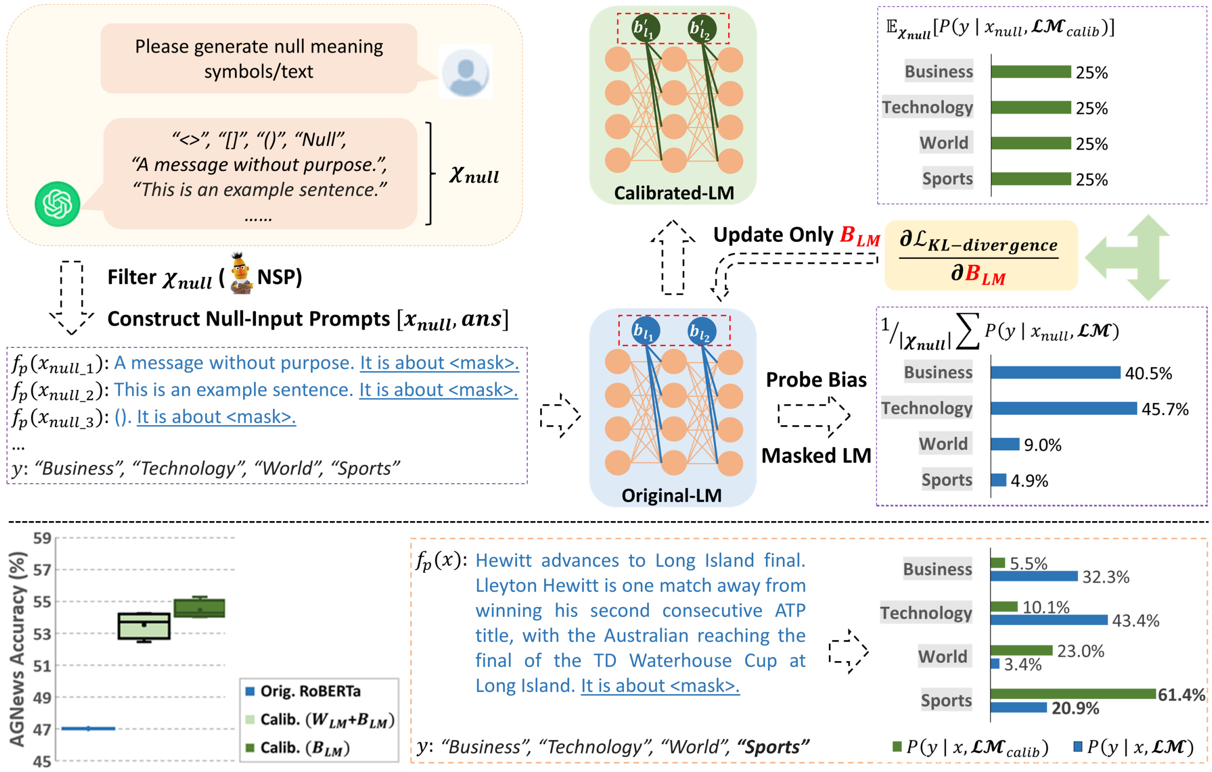

Figure 1: We demonstrate our calibration method significantly improves classification performance of pre-trained LM. **Upper**: The pipeline of proposed null-input prompting method for intrinsic bias calibration targeting AGNews task (Zhang et al., 2015). **Lower left**: Performance comparison of zero-shot 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.

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 understanding 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 (Kullback and Leibler, 1951) 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 Preliminaries

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*[1] (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 from related work. 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 (Garimella et al., 2021; Meade et al., 2022).

Efficiently calibrating intrinsic bias in pre-trained LMs for enhancing downstream zero/few-shot prompt learning performance is an open research problem. In this work, we introduce a parameter-efficient intrinsic-bias calibration method leveraging automatically constructed null-input prompts. We demonstrate its effectiveness of making LMs better zero/few-shot learners for both in-context learning and prompt-based fine-tuning.

[1]For example, in a restaurant review "*I had roast chicken and a salad.*", RoBERTa model classifies the sentiment for "*roast chicken*" as positive, while the true label is neutral. This may arise from the association of "roast chicken" with positive sentiment words prevalent in pre-training corpora.

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:

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

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 $\bar{P}_{\mathcal{X}_{\text{null}}}(y \mid x_{\text{null}}, \mathcal{LM})$ may exhibit significant difference attributed to model's intrinsic bias. Ideally, for bias-calibrated LM $\mathcal{LM}_{calib}$, the expectation of output distribution conditioned on null-meaning inputs should be uniform across all label words, i.e.,

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

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

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

where $U(\mathcal{Y})$ denotes uniform probability distribution and $\bar{P}_{\mathcal{X}_{\text{null}}}(y\,|\,x_{\text{null}})$ is simplified as $\bar{P}_{\mathcal{X}_{\text{null}}}(y)$.

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 $\boldsymbol{W_{LM}}$[2] 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:

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

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.

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

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, deviating from our intended objective. We develop different 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). As shown in Figure 2, we observe that model performance has steep increases in the first one/few calibration batches[3] 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 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.

For *few-shot* downstream tasks. With the acquisition of a few labeled data, the previous challenge of lacking validation for determining the stopping point in the calibration process is alleviated. We leverage the downstream training dataset $\mathcal{D}^{\text{downstrm}}_{\text{train}}$ constituting K samples per class as validation dataset $\mathcal{D}^{\text{calib}}_{\text{val}}$ in the calibration.

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 the validation dataset $\mathcal{D}^{\text{calib}}_{\text{val}}$ might fail to identify such stopping points. To this effect, we store both $LM^{\text{one\_batch}}_{\text{calib}}$ (obtained from one-batch stopping) and $LM^{\text{val}}_{\text{calib}}$ (obtained from validation-based stopping) for downstream few-shot leaning tasks. Since $LM^{\text{one\_batch}}_{\text{calib}}$ is stored

[3]We experimented with smaller calibration learning rates and observed consistent less improvement of LM performance.

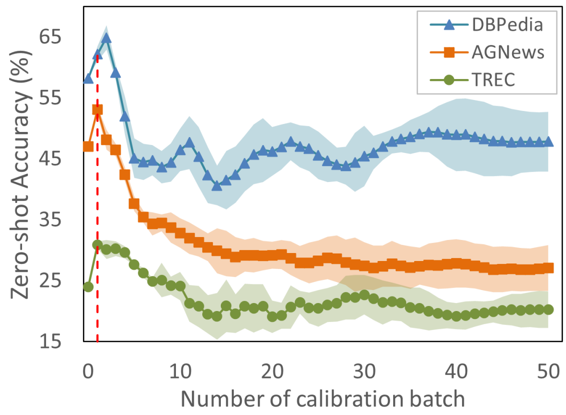

Figure 2: Empirical experiments show the impact of calibration on zero-shot learning performance as the number of calibration batches increases (batch size is 32). The intersections of the curves and red vertical line signify the outcomes of the first calibration batch.

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. However, creating numerous null-meaning inputs manually could be laborious and challenging. 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>."*. Note that this generation process is task-agnostic, ensuring that each generated data contains null information with respect to any downstream task.

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.

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
Password123	0.0369
123abc	0.0267
////////////////////	0.0008

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

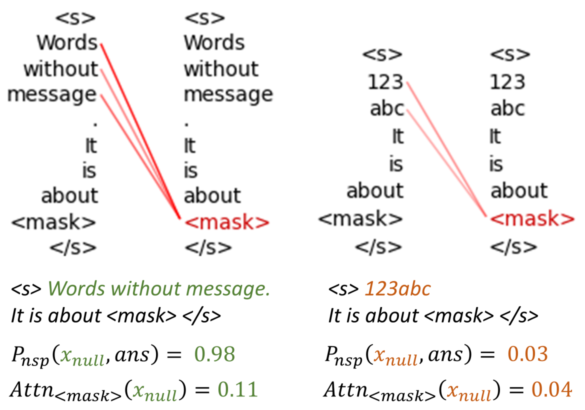

Figure 3: Visualization of attention score by the depth of color in the connecting lines. We only show the attention between `<mask>` token and null-meaning input x_{null}. $Attn_{<mask>}(x_{\text{null}})$ is the attention score of `<mask>` on x_{null}, averaged over encoder layers and attention heads. **Left**: Higher attention score indicates enhanced pattern extraction from x_{null} which has higher $P_{nsp}(x_{\text{null}}, ans)$.

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. We find this strategy brings advantages in two aspects: (1) It filters some null-meaning inputs that do not conform to normal symbols or text (e.g., *"////////////////////"*); (2) In the prompts, null-meaning inputs with higher $P_{nsp}(x_{\text{null}}, ans)$ exhibit higher attention scores between the null input and `<mask>` as shown in Figure 3. This indicates more effective conveyance of

	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	$47.0_{0.0}$	$54.3_{1.0}$	$\mathbf{54.5}_{0.6}$	$79.7_{0.8}$	$78.8_{3.3}$	$\mathbf{82.4}_{0.9}$	$\mathbf{89.1}_{0.9}$	$86.3_{1.6}$	$89.0_{0.8}$	$86.9_{2.8}$	$87.5_{1.3}$	$\mathbf{89.3}_{0.9}$
DBPedia	$58.2_{0.0}$	$54.1_{1.9}$	$\mathbf{61.8}_{0.6}$	$92.6_{0.6}$	$94.0_{0.9}$	$\mathbf{94.8}_{0.7}$	$98.2_{1.3}$	$99.0_{0.5}$	$\mathbf{99.0}_{0.1}$	$98.6_{0.3}$	$98.5_{0.2}$	$\mathbf{98.9}_{0.3}$
TREC	$24.0_{0.0}$	$29.4_{2.1}$	$\mathbf{31.1}_{0.5}$	$48.3_{1.4}$	$42.5_{3.4}$	$\mathbf{48.6}_{2.2}$	$85.0_{7.4}$	$82.2_{2.0}$	$\mathbf{89.3}_{4.5}$	$87.6_{2.5}$	$74.2_{4.0}$	$\mathbf{89.7}_{1.0}$
Subj	$50.8_{0.0}$	$\mathbf{64.0}_{2.7}$	$62.7_{0.8}$	$47.2_{0.2}$	$55.0_{1.3}$	$\mathbf{63.5}_{2.3}$	$91.2_{0.9}$	$88.2_{2.5}$	$\mathbf{93.2}_{1.2}$	$91.4_{3.3}$	$93.0_{0.8}$	$\mathbf{94.3}_{0.2}$
SST-5	$31.5_{0.0}$	$33.0_{2.1}$	$\mathbf{37.5}_{0.4}$	$34.4_{1.7}$	$31.2_{2.6}$	$\mathbf{36.6}_{1.0}$	$47.8_{4.6}$	$45.3_{2.8}$	$\mathbf{49.9}_{2.7}$	$47.1_{1.9}$	$42.6_{4.0}$	$\mathbf{50.0}_{1.7}$
Laptop	$54.6_{0.0}$	$58.3_{2.5}$	$\mathbf{59.6}_{1.9}$	$50.8_{1.0}$	$65.1_{2.7}$	$\mathbf{67.4}_{1.7}$	$74.3_{1.4}$	$74.3_{1.6}$	$\mathbf{74.9}_{2.9}$	$76.8_{1.0}$	$75.6_{1.4}$	$\mathbf{78.7}_{1.4}$
Restaurant	$68.6_{0.0}$	$72.0_{4.9}$	$\mathbf{72.8}_{1.6}$	$69.8_{1.1}$	$\mathbf{74.3}_{1.6}$	$74.0_{1.0}$	$79.7_{2.2}$	$79.0_{1.0}$	$\mathbf{82.0}_{0.9}$	$78.4_{4.9}$	$79.0_{5.5}$	$\mathbf{79.8}_{4.5}$
Twitter	$19.7_{0.0}$	$43.4_{4.1}$	$\mathbf{51.7}_{0.4}$	$21.0_{0.5}$	$40.7_{5.4}$	$\mathbf{49.4}_{2.7}$	$51.7_{2.9}$	$44.1_{3.9}$	$\mathbf{57.0}_{4.2}$	$57.7_{2.8}$	$50.3_{4.2}$	$\mathbf{59.3}_{2.3}$
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.

the *"null"* information to the placeholder <mask>, which could facilitate LM deciphering the *"null"* patterns of the prompts and benefit calibration.

After the sorting, we discard the bottom 20% x_{null} instances and obtain 800 null-meaning inputs. Specially, 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 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. 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 these aspect-level datasets, the task is to predict sentiments associated with the marked aspects in each sentence. More details are in Appendix A and 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.

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-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.

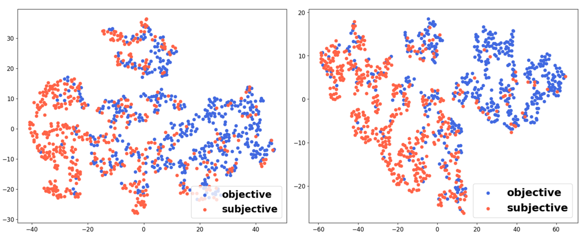

Figure 4: t-SNE visualization for output representations of <mask> token. **Left** is obtained from original LM; **Right** is obtained from the LM after *One-batch Calibration*. Two colors denote the two classes in Subj task.

The output representations of <mask> token for label word predictions are visualized by t-SNE (van der Maaten and Hinton, 2008) in Figure 4. 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 $P_{nsp}(x_{\text{null}}, ans)$. Note that for In-context lrn no demo (i.e., zero-shot learning in § 3.3), only batch size N of x_{null} are randomly sampled (*UnSel.*) or strategically selected (*Sel.*).

	In-context lrn no demo		**Prompt FT no demo**	
	$\boldsymbol{W_{LM}+B_{LM}}$	$\boldsymbol{B_{LM}}$	$\boldsymbol{W_{LM}+B_{LM}}$	$\boldsymbol{B_{LM}}$
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. $\boldsymbol{W_{LM}+B_{LM}}$ updates entire LM in calibration while $\boldsymbol{B_{LM}}$ only updates bias parameters.

5.3 Update Entire LM vs. Only Bias Parameters in Calibration

In Table 4, we evaluate the impact of updating entire LM ($\boldsymbol{W_{LM}+B_{LM}}$) during calibration on downstream task performance, as compared to only updating bias parameters ($\boldsymbol{B_{LM}}$) in calibration. The optimal learning rate for updating entire LM in calibration is smaller as shown in Appendix A Table 6. Results of In-context lrn/Prompt FT *with demo* are in Appendix B (Table 11). 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. 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 $\boldsymbol{B_{LM}}$ in calibration.

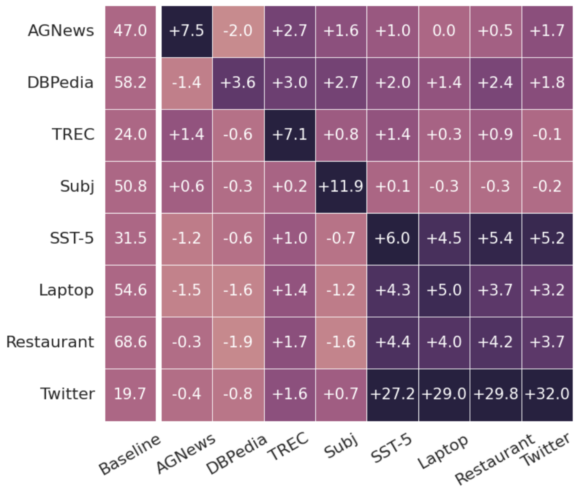

Figure 5: Impact of calibration on general modeling abilities shown through the changes with respect to baseline on each column. Each row shows the zero-shot performance of one task employing *original LM* (first column; baseline), *task-specific calibrated LM* (diagonal), *other-task calibrated LM* (other places).

5.4 Analysis

Evaluate language modeling abilities after calibration. 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 *general* language modeling abilities. Specifically, we take the LM calibrated for one task and evaluate its performance on the other tasks. The results are shown in Figure 5. In general, intrinsic bias calibration for one task has a minimal adverse effect on other tasks' modeling (no more than 2% degradation) due to 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.[4]

Is the improvement in prompt-based fine-tuning simply attributed to *NoisyTune*? Wu et al. (2022) demonstrates that adding noise to pre-trained LMs

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

benefits conventional fine-tuning on downstream tasks. To validate that the gains in prompt-based fine-tuning with our method are not solely a result of perturbing LM parameters, we conduct comparison experiments by adding noise (with the most suitable intensity suggested in Wu et al., 2022) to the pre-trained LM before initiating prompt-based fine-tuning. Table 5 illustrates that, while NoisyTune proves effective in better fine-tuning pre-trained LMs on downstream tasks (as compared to NoCal in Table 2), our method consistently surpasses NoisyTune, confirming the efficacy of intrinsic bias calibration in enhancing LM performance.

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.

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

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 they have demonstrated competitive performance (Gao et al., 2021; Logan IV et al., 2022). 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 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

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.

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.

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.

Aparna Garimella, Akhash Amarnath, Kiran Kumar, Akash Pramod Yalla, N Anandhavelu, Niyati Chhaya, and Balaji Vasan Srinivasan. 2021. He is very intelligent, she is very beautiful? on mitigating social biases in language modelling and generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4534–4545.

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*.

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*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

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.

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

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.

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.

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*.

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.

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.

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.

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}.

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 $lr_{calib} = \{1e-6, 1e-5, 1e-4, 1e-3\}$, 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		Prompt FT (downstream)
	$\boldsymbol{W_{LM}} + \boldsymbol{B_{LM}}$	$\boldsymbol{B_{LM}}$	
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*
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(P || U)    ▷ Unif. distribution U
    B_LM ← B_LM − α · ∂L/∂B_LM    ▷ Freeze W_LM
    if first batch then
        Save LM^{one_batch}_{calib}
    end if
    if downstream is zero_shot then break
    end if
    if better Compute_Metric(D^{calib}_{val}) then
        Save LM^{val}_{calib}
    end if
end for
```

B Additional Results

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.

	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	$37.8_{0.0}$	$36.2_{4.6}$	$\mathbf{49.0}_{0.9}$	$68.4_{0.4}$	$69.7_{4.3}$	$\mathbf{73.7}_{0.3}$	$88.2_{0.3}$	$87.8_{0.6}$	$\mathbf{88.9}_{1.0}$	$86.7_{0.1}$	$74.2_{4.1}$	$\mathbf{87.2}_{0.1}$
DBPedia	$\mathbf{57.2}_{0.0}$	$50.5_{7.1}$	$54.9_{0.1}$	$56.5_{3.4}$	$78.7_{4.4}$	$\mathbf{83.9}_{0.4}$	$95.2_{2.1}$	$93.5_{5.0}$	$\mathbf{99.0}_{0.4}$	$97.8_{0.9}$	$96.7_{0.8}$	$\mathbf{98.6}_{0.1}$
TREC	$28.2_{0.0}$	$25.4_{4.4}$	$\mathbf{30.2}_{0.1}$	$41.2_{0.3}$	$39.9_{3.8}$	$\mathbf{42.5}_{1.0}$	$82.5_{10.9}$	$70.3_{2.3}$	$\mathbf{86.4}_{6.5}$	$85.7_{1.8}$	$80.6_{5.0}$	$\mathbf{91.2}_{0.6}$
Subj	$53.6_{0.0}$	$63.6_{1.9}$	$\mathbf{66.4}_{1.8}$	$50.8_{0.2}$	$67.0_{1.7}$	$\mathbf{69.6}_{0.4}$	$\mathbf{92.5}_{1.3}$	$91.1_{0.4}$	$91.9_{1.7}$	$90.4_{2.1}$	$92.0_{0.2}$	$\mathbf{92.3}_{0.1}$
SST-5	$31.9_{0.0}$	$30.8_{3.4}$	$\mathbf{32.2}_{0.2}$	$25.3_{4.3}$	$28.6_{3.4}$	$\mathbf{29.8}_{1.7}$	$45.9_{3.3}$	$42.9_{2.3}$	$\mathbf{48.1}_{1.8}$	$44.3_{5.2}$	$40.7_{2.5}$	$\mathbf{45.8}_{2.6}$
Laptop	$56.1_{0.0}$	$56.7_{3.8}$	$\mathbf{60.0}_{0.1}$	$49.2_{0.9}$	$61.5_{2.8}$	$\mathbf{64.0}_{0.6}$	$75.8_{3.4}$	$73.0_{1.3}$	$\mathbf{76.3}_{1.8}$	$74.8_{0.1}$	$76.0_{0.6}$	$\mathbf{76.3}_{0.5}$
Restaurant	$69.8_{0.0}$	$\mathbf{72.0}_{2.9}$	$69.5_{0.5}$	$67.6_{0.7}$	$70.5_{2.4}$	$\mathbf{73.2}_{0.7}$	$75.5_{6.6}$	$\mathbf{77.3}_{3.4}$	$77.2_{1.1}$	$74.8_{3.3}$	$75.2_{0.7}$	$\mathbf{76.1}_{3.9}$
Twitter	$22.0_{0.0}$	$48.6_{5.1}$	$\mathbf{52.3}_{0.6}$	$17.6_{0.4}$	$41.8_{5.4}$	$\mathbf{48.4}_{0.5}$	$54.5_{1.1}$	$47.7_{3.8}$	$\mathbf{57.9}_{1.3}$	$50.6_{4.6}$	$51.8_{2.1}$	$\mathbf{56.0}_{4.9}$
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 8: 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	$70.4_{6.7}$	$\mathbf{76.3}_{3.6}$	$76.4_{5.4}$	$\mathbf{80.2}_{8.0}$	$78.2_{1.3}$	$\mathbf{83.2}_{1.1}$
	DBPedia	$92.9_{0.9}$	$\mathbf{94.0}_{1.0}$	$97.0_{1.6}$	$\mathbf{98.4}_{0.9}$	$97.4_{1.0}$	$\mathbf{97.8}_{1.1}$
	TREC	$49.8_{4.2}$	$\mathbf{50.5}_{4.0}$	$49.1_{22.6}$	$\mathbf{60.3}_{9.6}$	$65.2_{9.3}$	$\mathbf{66.1}_{9.3}$
	Subj	$49.4_{1.1}$	$\mathbf{56.2}_{3.9}$	$66.4_{5.4}$	$\mathbf{82.2}_{5.9}$	$72.3_{13.9}$	$\mathbf{81.5}_{13.2}$
4-shot	AGNews	$75.7_{3.9}$	$\mathbf{80.3}_{1.7}$	$85.4_{2.7}$	$\mathbf{87.3}_{1.3}$	$76.7_{13.1}$	$\mathbf{85.9}_{1.9}$
	DBPedia	$93.0_{0.4}$	$\mathbf{93.9}_{0.4}$	$97.2_{0.8}$	$\mathbf{97.9}_{1.1}$	$96.4_{1.5}$	$\mathbf{98.6}_{0.6}$
	TREC	$51.9_{2.6}$	$\mathbf{53.2}_{2.5}$	$64.5_{7.1}$	$\mathbf{67.6}_{6.7}$	$73.6_{8.5}$	$\mathbf{78.2}_{9.7}$
	Subj	$48.8_{2.2}$	$\mathbf{59.4}_{3.1}$	$81.4_{3.9}$	$\mathbf{88.5}_{3.2}$	$78.9_{9.3}$	$\mathbf{83.6}_{7.8}$
8-shot	AGNews	$79.6_{1.0}$	$\mathbf{82.4}_{1.6}$	$86.9_{1.9}$	$\mathbf{88.1}_{0.4}$	$85.5_{1.7}$	$\mathbf{88.0}_{1.4}$
	DBPedia	$92.9_{0.8}$	$\mathbf{94.2}_{0.2}$	$97.3_{1.2}$	$\mathbf{98.8}_{0.5}$	$98.2_{0.8}$	$\mathbf{98.6}_{0.2}$
	TREC	$47.9_{2.2}$	$\mathbf{48.7}_{2.0}$	$71.6_{4.9}$	$\mathbf{72.2}_{5.1}$	$75.4_{6.2}$	$\mathbf{81.7}_{5.6}$
	Subj	$48.4_{1.0}$	$\mathbf{60.5}_{4.8}$	$91.9_{1.3}$	$\mathbf{92.7}_{0.8}$	$88.9_{5.3}$	$\mathbf{92.1}_{2.2}$

Table 9: 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).

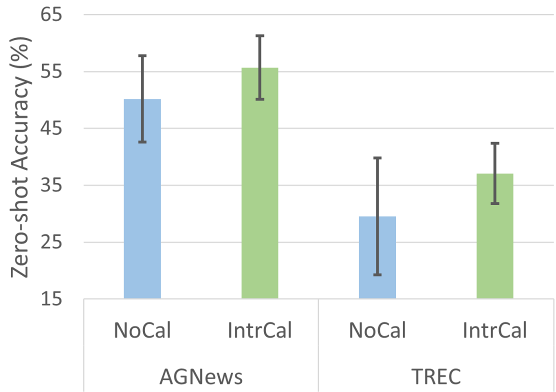

Figure 6: Performance comparison averaged on using five different prompt templates with RoBERTa-large. IntrCal (ours; intrinsic-bias calibrated LM) demonstrates significantly improved accuracy with lower variance compared to NoCal (no calibration).

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 10: The five different prompt templates used in Figure 6.

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

	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 12: 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.