Batch Calibration: Rethinking Calibration for In-Context Learning and Prompt Engineering

Han Zhou[1,2,*] **Xingchen Wan**[1] **Lev Proleev**[1] **Diana Mincu**[1]
Jilin Chen[1] **Katherine Heller**[1] **Subhrajit Roy**[1]
[1]Google Research [2]University of Cambridge
{hzhouml,xingchenw,levp,dmincu,jilinc,kheller,subhrajitroy}@google.com

Abstract

Prompting and in-context learning (ICL) have become efficient learning paradigms for large language models (LLMs). However, LLMs suffer from prompt brittleness and various bias factors in the prompt, including but not limited to the formatting, the choice verbalizers, and the ICL examples. To address this problem that results in unexpected performance degradation, calibration methods have been developed to mitigate the effects of these biases while recovering LLM performance. In this work, we first conduct a systematic analysis of the existing calibration methods, where we both provide a unified view and reveal the failure cases. Inspired by these analyses, we propose *Batch Calibration* (BC), a simple yet intuitive method that controls the contextual bias from the batched input, unifies various prior approaches, and effectively addresses the aforementioned issues. BC is zero-shot, inference-only, and incurs negligible additional costs. We validate the effectiveness of BC with PaLM 2-(S, M, L) and CLIP models and demonstrate state-of-the-art performance over previous calibration baselines across more than 10 natural language understanding tasks.

1 Introduction

Prompting large language models (LLMs) [4, 2] has become an efficient learning paradigm for adapting LLMs to a new task by conditioning on human-designed instructions. The remarkable in-context learning (ICL) ability of LLMs also leads to efficient few-shot learners that can generalize from few-shot input-label pairs [3, 27]. However, the predictions of LLMs are highly sensitive and even biased to the choice of templates [32], verbalizers [17], and demonstrations [26], resulting in barriers for pursuing efficiently adaptable and robust LLM applications. Extensive research has been devoted to mitigating these biases, which we explicitly refer to the a-priori propensity of LLMs to predict certain classes over others unfairly. Lu et al. [29] provide an analysis of the impacts of the order of ICL examples to LLMs and have explored the order selection mechanisms for ICL. On the other hand, Zhao et al. [69] reveal the bias of language models toward certain answers and propose to calibrate the LLM given content-free tokens.

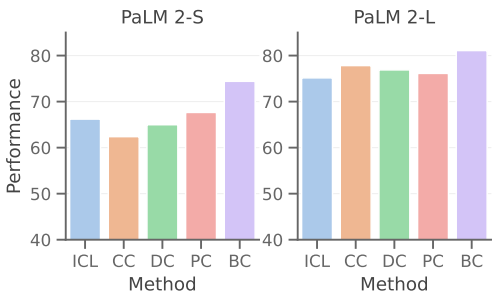

Figure 1: Batch Calibration (BC) achieves the best performance on 1-shot ICL over calibration baselines on an average of 13 classification tasks on PaLM 2-S and PaLM 2-L [2].

[*]Work done as a Student Researcher at Google.

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023.

More recently, Fei et al. [12] detect the domain-label bias, and Han et al. [14] treat the calibration of LLMs as learning a robust decision boundary. Though multiple calibration solutions have been provided, the field currently lacks a unified analysis that systematically distinguishes and explains the unique characteristics and downsides of each approach.

In this work, we first conduct a comprehensive analysis across existing calibration methods for LLMs. We approach the calibration problem from a distinctive point of view by interpreting the decision boundaries for each calibration method together with the ICL decision boundary. We start observing fatal failure cases for each method by extending them to more challenging and under-explored evaluation tasks. We then conclude the current limitation for each method with a novel interpretation from the decision boundary perspective, pointing to the need for a unified and widely applicable solution for conquering diverse bias sources in the field of LLM efficient learning.

Inspired by these findings, we propose *Batch Calibration* (BC), a zero-shot and inference-only calibration method for prompting and ICL. The central objective of BC is to accurately model the bias from the prompt context (referred to as *contextual bias* in this paper) by marginalizing the LLM scores in the batched input. The simplicity of the design of BC only brings negligible computation overhead at the output of the LLM. We conducted extensive experiments on more than 10 natural language understanding tasks. BC stands as the most widely applicable calibration method while achieving state-of-the-art results. We provide further analysis with BC on robustness with templates, ICL choices and orders, and verbalizers, validating that BC can effectively alleviate prompt brittleness and make prompt engineering easier. To summarize, we provide the following contributions:

- We provide a unified and systematic analysis of existing calibration methods through their decision boundaries, investigate the common use of content-free tokens as an estimator of contextual bias, and identify their deficiency with individual case studies.
- We propose Batch Calibration (BC), a zero-shot and inference-only calibration method for ICL, that mitigates the bias from the batch.
- We show that while conceptually simple, BC attains state-of-the-art performance in both zero-shot and few-shot learning setups over widely selected tasks with PaLM-2 and CLIP models.

2 A Systematic Analysis of Calibration

Bias in Prompting and In-Context Learning (ICL) Prompting is an efficient learning paradigm that allows LLMs to perform zero-shot inference by conditioning on a human-designed instruction. Formally, denoting a test query-target pair $\{x_i, y_i\}$ and instruction as the context C for a classification task, LLMs make prediction by computing: $\arg\max_{y \in \mathcal{Y}} \mathbf{p}(y|x_i, C)$, where $\mathbf{p} \in \mathbb{R}^J$ are the logits, and $\mathcal{Y}$ denotes the verbalizers that define the label set for J classes. ICL further enables LLM to learn from k input-label pairs (i.e., few-shot setup), $s^{(i)} = \texttt{Template}(x^{(i)}, y^{(i)}) \, \forall i \in \{1, ..., k\}$, by concatenating few-shot demonstrations in a pre-defined template as the context, $C = \texttt{Concat}(s^{(i)}, ..., s^{(k)})$. Though ICL has demonstrated strong performance with easy implementations, the prediction of LLMs is shown to be biased towards certain answers due to different elements of $\mathbf{p}(y|x_i, C)$ [29]. In the ICL context C, majority label bias and recency label bias [69] can bias the prediction of LLMs toward the most frequent label and the label towards the end of the demonstration, respectively. Among verbalizer tokens $y_j \in \mathcal{Y}$, LLMs are shown to be inherently biased towards predicting the label-tokens that appear more frequently from pretraining term statistics [48, 44]. These bias factors significantly degrade the performance of LLMs for robust ICL applications.

Overview of ICL Calibration Methods. *Contextual Calibration* [69] (CC): Motivated by a common calibration technique that applies affine transformation on the model outputs [41, 13], Zhao et al. [69] propose to calibrate the LLM prediction by first measuring the entire test-time distribution $\hat{\mathbf{p}}$ by a content-free input. Using "`N/A`" as a content-free example, the model score distribution is generated by $\hat{\mathbf{p}}_{\text{cf}} := \mathbf{p}(y|[\texttt{N/A}], C)$. CC then generates the calibrated output by transforming the uncalibrated scores $\mathbf{p}(y|x, C)$ with $\mathbf{W} \in \mathbb{R}^{J \times J}$ via $\mathbf{W}\mathbf{p}(y|x, C)$, where $\mathbf{W} = \text{diag}(\hat{\mathbf{p}}_{\text{cf}})^{-1}$ offsets the uncalibrated scores with the model score (a contextual prior) triggered by the content-free sample.

Domain-Context Calibration [12] (DC): Instead of using a single content-free token, Fei et al. [12] propose DC that estimates a contextual prior $\hat{\mathbf{p}}(y|C)$ by using a random in-domain sequence.

Table 1: Calibration methods with their mathematical formulation and their equivalent decision boundary derivations in a two-dimensional problem. The cost for the number of API calls is denoted as #Forward, where 1 counts for the ICL forward cost. The potential failure case for each calibration method in practical scenarios is marked as ✗.

Method	Token	#Forward	Comp. Cost	Cali. Form	Learning Term	Decision Boundary $h(\mathbf{p})$	Multi-Sentence	Multi-Class
CC	N/A	$1+1$	Inverse	$\mathbf{Wp}+\mathbf{b}$	$\mathbf{W}=\text{diag}(\hat{\mathbf{p}})^{-1}, \mathbf{b}=\mathbf{0}$	$p_0=\alpha p_1$	✗	✓
DC	Random	$20+1$	Add	$\mathbf{Wp}+\mathbf{b}$	$\mathbf{W}=\mathbf{I}, \mathbf{b}=-\frac{1}{T}\sum_t \mathbf{p}(y\|\text{text}_j, C)$	$p_0=p_1+\alpha$	✗	✓
PC	-	1	EM-GMM	-	$\sum_j \alpha_j P_G(\mathbf{p}\|\boldsymbol{\mu}_\mathbf{j}, \boldsymbol{\Sigma}_\mathbf{j})$	$P_\text{G}(\mathbf{p}\|\mu_0,\Sigma_0)=P_\text{G}(\mathbf{p}\|\mu_1,\Sigma_1)$	✓	✗
BC (Ours)	-	1	Add	$\mathbf{Wp}+\mathbf{b}$	$\mathbf{W}=\mathbf{I}, \mathbf{b}=-\mathbb{E}_x\left[\mathbf{p}(y\|x,C)\right]$	$p_0=p_1+\alpha$	✓	✓

It randomly sampled L tokens at an average sentence length from an unlabeled text set. Then, it estimates the content-free prediction prior by averaging the model score T times, such that: $\hat{\mathbf{p}}_\text{random}=\frac{1}{T}\sum_{t=1}^{T}\mathbf{p}(y|[\text{RANDOM TEXT}]_t, C)$. The final test-time prediction is then calibrated by dividing the estimated prior prediction, or equivalently in logits space, $\mathbf{p}(y|x_i,C)-\hat{\mathbf{p}}_\text{random}$.

Prototypical Calibration [14] (PC): PC learns a decision boundary with Gaussian mixture models (GMMs). It estimates J prototypical clusters for the model output $\mathbf{p}$ for J classes: $P_\text{GMM}(\mathbf{p})=\sum_{j=0}^{J-1}\alpha_j P_G(\mathbf{p}|\boldsymbol{\mu}_\mathbf{j}, \boldsymbol{\Sigma}_\mathbf{j})$, where P_G denotes a multi-variate Gaussian distribution, and the parameters: mixing coefficient α, mean vector $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$ are estimated by the Expectation-Maximization [34]. Followed by an automatic label assignment strategy, the predicted label is then computed by $\arg\max_j P_G(\mathbf{p}_j|\mu^*,\boldsymbol{\Sigma}^*)$ in the inference time. This EM-GMM process can require up to T repetitions to stabilize its estimation of clusters where T is a hyperparameter of the algorithm.

Summarizing the calibration methods with distinctive design principles discussed so far, in Table 1, we present a unified view of the characteristics of each method. Though each approach demonstrates a clear motivation for calibrating ICL, it is still unclear which method surpasses others in what scenarios. We proceed with an in-depth analysis of existing methods in representative tasks. In pursuit of practical guidelines for ICL calibration, we set out two important research questions behind their design principles: **1)** What constitutes a better decision boundary for calibrations? **2)** Is content-free prior a good estimator of contextual bias?

What Constitutes a Better Decision Boundary for Calibrations? To address this research question, we first derive the decision boundary for each category of calibration methods. We recall that the classification by a LLM is based on $\arg\max_{j\in\{0,\ldots,J-1\}} p_j$ where p_j denotes the j-th element of output vector $\mathbf{p}$. Consider binary classification problem for simplicity: the decision boundary $h(\mathbf{p})$ for ICL is given by the line $p_0-p_1=0$: the model predicts class 0, y_0, if $p_0-p_1\geq 0$, and class 1 otherwise. Consequently, CC and DC that apply an affine transformation at $\mathbf{p}$ is equivalent to a linear transformation to the decision boundary. In CC with $\mathbf{W}=\text{diag}(\hat{\mathbf{p}})^{-1}$, $\mathbf{b}=\mathbf{0}$, the decision boundary can then be derived as: $p_0\times\frac{1}{\hat{p}_0}=p_1\times\frac{1}{\hat{p}_1}\rightarrow p_0-p_1\times\frac{\hat{p}_0}{\hat{p}_1}=0$, which is a *rotation* of the ICL's linear decision boundary around the origin. Similarly, DC with $\mathbf{W}=\mathbf{I}$, $\mathbf{b}=-\frac{1}{T}\sum_t\mathbf{p}(y|[\text{RANDOM TEXT}]_t,C)=-\hat{\mathbf{p}}$ is equivalent to a *shift* of ICL's linear decision boundary away from the origin, such that $p_0-p_1=(\hat{p}_0-\hat{p}_1)$. It is worth noting that both calibration choices lead to a linear decision boundary, indicating that the calibration problem can be framed as an unsupervised decision boundary learning problem. Based on this intuition, we further derive the decision boundary for PC as $P_\text{G}(\mathbf{p}|\mu_0,\Sigma_0)-P_\text{G}(\mathbf{p}|\mu_1,\Sigma_1)=0$, which delivers a non-linear boundary between the estimated Gaussian mixtures. We conduct a preliminary experiment to visualize the derived decision bounds from existing calibration methods alongside the ICL baseline. In Fig. 2, we observe that uncalibrated ICL is biased towards predicting `negative` in the SST-2 task. This biased prediction is then mitigated by each calibration method, where we observe a rotated decision boundary from CC, a shifted boundary from DC, and a non-linear boundary between the GMMs by PC. However, in the QNLI task (bottom row of Fig. 2), we observe failure cases in the calibration baselines, in particular, PC (third figure from the left), where it fails to capture the correct distribution for each class. From Fig. 2 and the additional results in Fig. 9 in Appendix §E, we find that while theoretically more flexible, the non-linear decision boundaries learned by PC tend to be susceptible to overfitting and may suffer from instability in EM-GMM. We hypothesize that the PC boundary is even more vulnerable to instability for more challenging multi-class tasks due to the increased difficulties of learning clusters and assigning classes correctly. Conversely, we find that linear decision

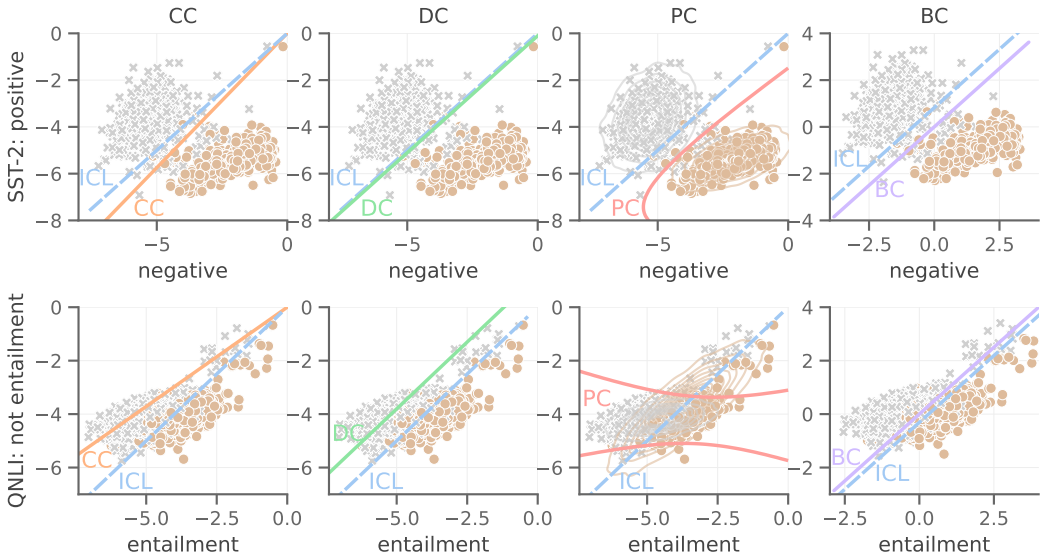

Figure 2: Visualization of the decision boundaries of uncalibrated ICL, and after applying existing calibration methods and the proposed BC (to be introduced in Sec 3) in representative binary classification tasks of SST-2 (top row) [51] and QNLI (bottom row) [56] on 1-shot PaLM 2-S. We show success and failure cases for each baseline method (CC, DC, and PC), whereas BC is consistently effective. Refer to Appendix §E for more examples.

boundaries, as evidenced by CC and DC, can be more robust and generalizable across tasks. We validate this hypothesis by proposing BC with extensive experiments in Sec. 4.

Is Content-free Input a Good Estimator of the Contextual Prior? CC and DC both use a linear decision boundary but differ from each other by leveraging different formats of a content-free input to estimate the contextual prior. However, as we observed in Fig. 2, they both exhibit failure cases in QNLI, a question-answering NLI task. We hypothesize that contrary to the proposals made by CC and DC, relying on content-free tokens for calibration is *not* always optimal and may even introduce additional bias, depending on the task type. For example, in a textual entailment task involving question-sentence pairs, we empirically observe that an ICL template employed with a content-free token '`N/A`' such as '`Question: N/A, Sentence: N/A, Answer:`' will result in a biased prediction towards '`entailment`', because although '`N/A`' is intended to be content-free, the LLM may nevertheless construe '`N/A`' in the sentence to be substantively entailed to the '`N/A`' in the question due to surface text equivalency. This phenomenon holds true for other multi-text classification tasks, such as paraphrasing tasks. Consequently, the prior estimated via a single content-free token can lead to further bias. DC introduces multiple randomly sampled tokens to form a content-free input, e.g. '`Question: that What old rubisco's the did Which?`'. We suspect a possible reason is that random sequences, when used in conjunction with in-context demonstrations, can be susceptible to spurious relations among them that often lead to unfair priors further skewing the predictions, which is also reflected in Fig. 2, where CC and DC fail to mitigate the contextual bias in the QNLI task. In sum, the empirical observation shows that content-free inputs can be inappropriate prior estimators, especially for multi-sentence classification tasks.

3 Batch Calibration

Batch Calibration (BC). Following the discussion in Sec. 2, we argue that the most critical component for calibration is to accurately estimate the contextual bias term $\mathbf{p}(y|C)$. Both CC and DC, which use content-free and in-domain random tokens as trigger signals, respectively, have failure cases in multi-sentence classification when the estimation of the contextual bias is inaccurate. On the other hand, PC is vulnerable to overfitting and may incorrectly model the mixtures, especially

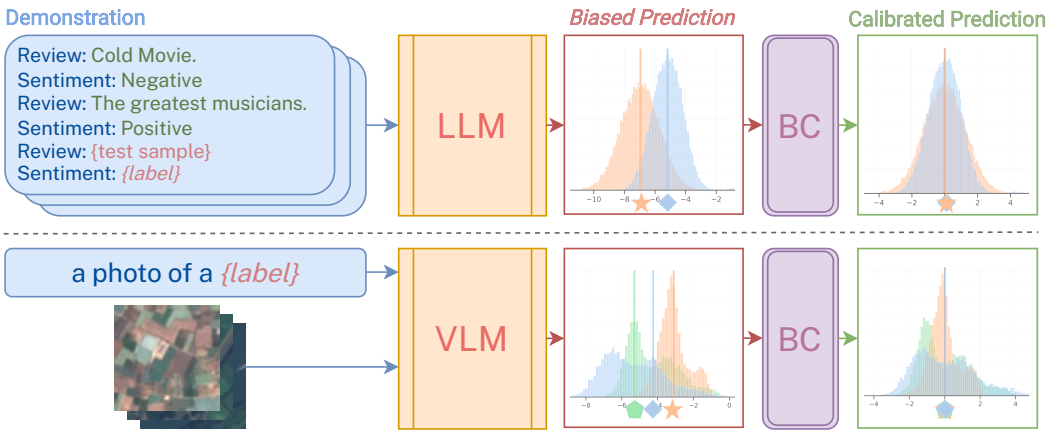

Figure 3: Illustration of Batch Calibration (BC). Batches of demonstrations with in-context examples and test samples are passed into the LLM. Due to implicit bias sources in the context, the score distribution from the LLM becomes highly biased. BC is a modular and adaptable layer option appended to the output of the LLM/VLM. BC generates calibrated scores according to Eq. 1 & 2. Highlighted symbols indicate the distribution means (visualized *for illustration only*).

in high-dimensional space. We, therefore, opt for a linear decision boundary for its robustness, and instead of relying on content-free tokens, we propose to estimate the contextual bias for each class $\mathbf{p}(y = y_j|C)$ from a batch with M samples, $\{x^1, ..., x^M\}$, in a *content-based* manner by marginalizing the output score over all samples $x \sim P(x)$ within the batch:

$$\mathbf{p}(y = y_j|C) = \mathop{\mathbb{E}}_{x \sim P(x)} \Big[\mathbf{p}(y = y_j|x, C)\Big] \approx \frac{1}{M} \sum_{i=1}^{M} \mathbf{p}(y = y_j|x^{(i)}, C) \, \forall \, y_j \in \mathcal{Y}. \tag{1}$$

We then obtain the calibrated probability by dividing the output probability over the contextual prior, which is equivalently by shifting the log-probability by the estimated mean of each class:

$$\hat{y}_i = \arg\max_{y \in \mathcal{Y}} \mathbf{p}_{\text{BC}}(y|x_i, C) = \arg\max_{y \in \mathcal{Y}} \big[\mathbf{p}(y|x_i, C) - \hat{\mathbf{p}}(y|C)\big]. \tag{2}$$

It is noteworthy that this BC procedure is zero-shot and only involves unlabeled test samples. BC incurs negligible computation costs. We may either compute the correction term $\hat{\mathbf{p}}(y|C)$ once all test samples are seen or, alternatively, in an on-the-fly manner that dynamically processes the outputs. To do so, we may use a running estimate of the contextual bias for BC. At the $n+1$ mini-batch, the bias term is given by: $\mathbf{p}_{\text{r}}^{n+1}(y|C) = \frac{n}{n+1}\mathbf{p}_{\text{r}}^{n}(y|C) + \frac{1}{n+1}\hat{\mathbf{p}}^{n+1}(y|C)$, thereby allowing BC's calibration term to be estimated from a small number of mini-batches that is subsequently stabilized when more mini-batches arrive.

4 Experiments

Experiments on Natural Language Tasks. We present the results across a diverse set of NLP tasks in Table 2. Notably, BC consistently outperforms ICL, yielding significant performance enhancement of 8% and 6% on PaLM 2-S and PaLM 2-L, respectively. This shows that the BC implementation successfully mitigates the contextual bias from the in-context examples and unleashes the full potential of LLM in efficient learning and quick adaptation to new tasks. In addition, BC improves over the state-of-the-art PC baseline by 6% on PaLM 2-S, and surpasses the competitive CC baseline by another 3% on average on PaLM 2-L. Specifically, BC is a generalizable and cheaper technique across all evaluated tasks, delivering stable performance improvement, whereas previous baselines exhibit varying degrees of instability across tasks: DC baseline is the least competitive; CC displays more failure cases in multi-sentence classification tasks, particularly for paraphrasing and NLI tasks, as we hypothesized in Sec 2; PC, while occasionally competitive, exhibits large performance fluctuations, as evidenced by its large standard deviation, resulting in frequent substantial performance degradation. We attach vision-language model results in Appendix §E.

Table 2: Accuracy (%) on natural language classification tasks with 1-shot PaLM 2-S and PaLM 2-L Models [2]. We report the mean and standard deviation for all results for 5 different in-context examples. We reproduce all baselines, and the implementation details are described in Appendix §D. The **best** and **second-best** results are marked in bold fonts and ranked by color.

Model	PaLM 2-S					PaLM 2-L				
Method	ICL	CC	DC	PC	BC	ICL	CC	DC	PC	BC
SST-2	$93.62_{0.62}$	**$95.50_{0.25}$**	$94.29_{0.32}$	**$95.71_{0.10}$**	$95.44_{0.15}$	$93.16_{5.18}$	**$95.82_{0.62}$**	$94.91_{2.01}$	$95.64_{0.47}$	**$95.78_{0.55}$**
MNLI	**$68.52_{7.98}$**	$60.07_{11.26}$	$63.45_{1.99}$	$59.29_{13.79}$	**$75.12_{2.76}$**	$72.77_{3.65}$	**$79.45_{3.46}$**	$71.53_{4.86}$	$78.68_{7.10}$	**$81.34_{2.29}$**
QNLI	**$81.20_{1.90}$**	$56.86_{3.29}$	$65.62_{3.53}$	$69.82_{17.73}$	**$82.45_{1.82}$**	$64.68_{3.53}$	**$69.71_{4.89}$**	$68.97_{3.27}$	$61.01_{15.26}$	**$87.90_{1.24}$**
MRPC	$66.42_{10.15}$	**$70.44_{0.94}$**	$68.58_{0.21}$	**$71.86_{1.29}$**	$70.05_{2.40}$	**$73.19_{1.21}$**	$72.40_{3.53}$	$68.68_{0.40}$	**$75.39_{2.60}$**	$70.39_{2.56}$
QQP	$63.91_{0.66}$	**$65.55_{5.34}$**	$53.92_{9.35}$	$65.28_{3.42}$	**$71.48_{1.46}$**	**$82.57_{0.75}$**	$81.17_{2.03}$	$78.32_{1.82}$	**$81.42_{0.24}$**	$79.56_{1.40}$
BoolQ	$83.99_{3.90}$	$87.14_{1.60}$	$87.64_{1.10}$	**$88.70_{0.15}$**	**$87.83_{0.10}$**	$90.02_{0.60}$	**$90.15_{0.54}$**	$87.77_{1.17}$	$64.40_{22.37}$	**$90.10_{0.22}$**
CB	$45.71_{10.61}$	$29.64_{7.85}$	$65.71_{3.20}$	**$81.07_{9.42}$**	**$78.21_{3.19}$**	**$92.86_{2.19}$**	$85.72_{7.78}$	**$92.86_{2.82}$**	$89.29_{7.25}$	**$93.21_{1.49}$**
COPA	**$96.40_{2.30}$**	$95.80_{2.05}$	**$96.40_{2.88}$**	**$96.20_{2.05}$**	**$96.40_{2.07}$**	**$98.60_{1.14}$**	$97.20_{1.10}$	$97.40_{0.89}$	**$99.00_{0.71}$**	$97.00_{1.00}$
RTE	**$80.94_{1.29}$**	$79.78_{0.92}$	$76.82_{1.72}$	$80.43_{1.07}$	**$83.47_{1.10}$**	$75.09_{2.11}$	$80.00_{2.48}$	$79.21_{1.95}$	**$86.64_{2.62}$**	**$85.42_{2.48}$**
WiC	$50.69_{0.59}$	$50.56_{0.50}$	$49.97_{0.13}$	**$51.38_{3.56}$**	**$61.10_{2.07}$**	$51.35_{1.90}$	$55.58_{6.38}$	$54.67_{6.02}$	**$57.87_{11.08}$**	**$64.83_{8.59}$**
ANLI-R1	**$46.24_{4.21}$**	$42.54_{3.20}$	$40.26_{3.66}$	$40.28_{6.46}$	**$59.82_{0.51}$**	$63.06_{2.63}$	$71.92_{3.71}$	**$73.56_{3.88}$**	$72.30_{8.05}$	**$75.00_{3.03}$**
ANLI-R2	$40.44_{0.90}$	$38.36_{0.82}$	$38.44_{3.46}$	**$41.88_{4.50}$**	**$50.16_{0.82}$**	$58.40_{1.19}$	$65.36_{3.75}$	**$65.48_{1.91}$**	$64.98_{2.94}$	**$67.30_{2.34}$**
ANLI-R3	$42.53_{0.99}$	$38.78_{1.04}$	**$43.67_{5.25}$**	$37.50_{0.81}$	**$55.75_{1.66}$**	$61.35_{3.14}$	**$67.32_{0.98}$**	$66.23_{0.72}$	$63.03_{6.03}$	**$66.38_{0.74}$**
Avg.	66.20	62.39	64.98	**67.65**	**74.41**	75.16	**77.83**	76.89	76.13	**81.09**

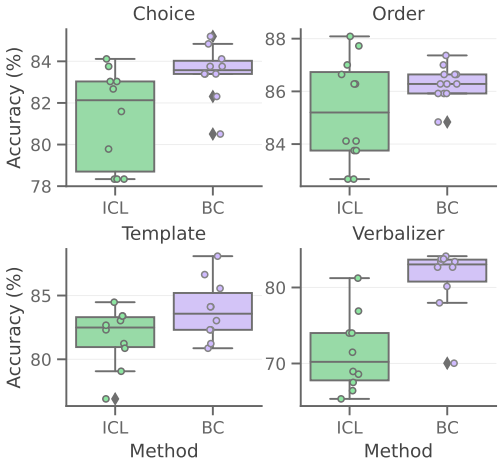

Figure 4: *BC makes prompt engineering easier:* Performance of BC with respect to ICL choices, ICL orders, prompt templates, and verbalizers.

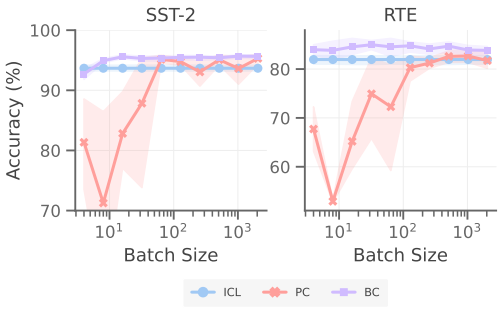

Figure 5: *BC is data-efficient and insensitive to the batch size:* Performance of BC across different sizes of an initial unlabeled set without using a running estimate of the contextual bias. We compare BC with the state-of-the-art PC baseline that also leverages unlabeled estimate set, and experiments are conducted on PaLM 2-S.

Robustness and Ablation Studies. We analyze the robustness of BC with respect to common prompt engineering design choices that were previously shown to significantly affect LLM performance [29, 26]: choices and orders of in-context examples, the prompt template for ICL, and the verbalizers, as shown in Fig. 4 evaluated on RTE. Setup details are listed in Appendix §F. First, we find that BC is more robust to ICL choices and can mostly achieve the same performance with different ICL examples. Additionally, given a single set of ICL shots, altering the order between each ICL example has minimal impact on the BC performance. However, it is worth noting that an optimal order selection can still lead to promising ICL performance. Furthermore, we analyze the robustness of BC under 10 designs of prompt templates, where BC shows consistent improvement over the ICL baseline. Therefore, though BC makes further improvements, a well-designed template can further enhance the performance of BC. Lastly, we examine the robustness of BC to variations in verbalizer designs. Remarkably, even when employing unconventional choices such as emoji pairs as the verbalizers leading to dramatic oscillations of ICL performance, BC largely recovers performance. This observation shows BC robustifies LLM predictions under common prompt design choices.

Batch Size. We study the impact of batch size on the performance of BC as shown in Fig. 5. In contrast to PC, which also leverages an unlabeled estimate set, BC is remarkably more sample

efficient, achieving a strong performance with only around 10 unlabeled samples, whereas PC requires more than 500 unlabeled samples before its performance stabilizes.

5 Related Work

Understanding and Improving ICL. Lu et al. [29] show the sensitivity of LLMs to ICL examples. This phenomenon is further explained through the effect from pretraining term frequencies [44] and corpora [48]. Meanwhile, Xie et al. [62] explains the ICL process through implicit Bayesian inference, and Wei et al. [60] show the emergent ability of LLMs by learning new input-label mappings. Various methods have been proposed to optimally select better in-context templates [53, 38, 65] and examples [46, 26, 55]. Specifically, Wan et al. [54] introduce a selection criteria based on the consistency, diversity, and repetition of in-context examples. Recently, noisy channel prompting [31] and flipped learning [64] have been proposed for robust ICL. Learning to assign labels by k-nearest neighbors [63] and training decoder networks [7] are also effective alternatives for few-shot ICL.

Bias in ICL and Calibrating LLMs. Zhao et al. [69] reveal the instability of LLMs in few-shot learning and demonstrate three bias sources: majority label bias, recency bias, and common token bias, as the bias factors behind the instability. They propose contextual calibration (CC) to mitigate these biases by grounding the prediction based on a content-free token as sample inputs. Si et al. [50] characterize the feature bias of LLMs, and Wang et al. [59] introduce the positional bias in candidate choices. Fei et al. [12] further observe the existence of domain-label bias and propose domain-context calibration (DC) that uses random in-domain tokens for estimating the bias. Meanwhile, Han et al. [14] analyze the impact of decision boundary for text classification tasks and propose to estimate prototypical clusters by Gaussian mixture models, thereby learning a robust decision boundary. Concurrently with our work, Pezeshkpour & Hruschka [39] spot the positional bias in multiple-choice questions, and Zheng et al. [70] propose to debias the positional bias in multiple choices with permutation-based prior estimation. BC differentiates from these methods as a generalizable solution across challenging classification tasks and modalities.

6 Conclusion

We first revisit previous calibration methods while addressing two critical research questions from an interpretation of decision boundaries, revealing their failure cases and deficiencies. We then propose Batch Calibration, a zero-shot and inference-only calibration technique. While methodologically simple and easy to implement with negligible computation cost, we show that BC scales from a language-only setup to the vision-language context, achieving state-of-the-art performance in both modalities. BC significantly improves the robustness of LLMs with respect to prompt designs, and we expect easy prompt engineering with BC while exploring the potential of BC to generative tasks in the future.

Acknowledgement

We thank Emily Salkey for her sincere project management support. We also thank Mohammad Havaei, Chirag Nagpal, Stephen Pfohl, Alexander D'Amour, and Ahmad Beirami for fruitful suggestions and feedbacks and the PaLM 2 team at Google for helping with occasional infrastructure questions.

References

[1] Alayrac, J., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J. L., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[4] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[5] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pp. 3606–3613, 2014.

[6] Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[7] Cui, G., Li, W., Ding, N., Huang, L., Liu, Z., and Sun, M. Decoder tuning: Efficient language understanding as decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15072–15087, Toronto, Canada, July 2023. Association for Computational Linguistics.

[8] De Marneffe, M.-C., Simons, M., and Tonhauser, J. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pp. 107–124, 2019.

[9] Deng, M., Wang, J., Hsieh, C.-P., Wang, Y., Guo, H., Shu, T., Song, M., Xing, E., and Hu, Z. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3369–3391, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[10] Diao, S., Huang, Z., Xu, R., Li, X., Yong, L., Zhou, X., and Zhang, T. Black-box prompt learning for pre-trained language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[11] Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[12] Fei, Y., Hou, Y., Chen, Z., and Bosselut, A. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14014–14031, Toronto, Canada, July 2023. Association for Computational Linguistics.

[13] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1321–1330, 2017.

[14] Han, Z., Hao, Y., Dong, L., Sun, Y., and Wei, F. Prototypical calibration for few-shot learning of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[16] Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[17] Holtzman, A., West, P., Shwartz, V., Choi, Y., and Zettlemoyer, L. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[18] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 2790–2799, 2019.

[19] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[20] Hu, X., Uzunbas, G., Chen, S., Wang, R., Shah, A., Nevatia, R., and Lim, S.-N. Mixnorm: Test-time adaptation through online normalization estimation. *arXiv preprint arXiv:2110.11478*, 2021.

[21] Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 1988–1997, 2017.

[22] Khurana, A., Paul, S., Rai, P., Biswas, S., and Aggarwal, G. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*, 2021.

[23] Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[24] Li, X. L. and Liang, P. 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)*, pp. 4582–4597, Online, August 2021. Association for Computational Linguistics.

[25] Lim, H., Kim, B., Choo, J., and Choi, S. TTN: A domain-shift aware batch normalization in test-time adaptation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[26] Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics.

[27] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[28] Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[29] Lu, Y., Bartolo, M., Moore, A., Riedel, S., and Stenetorp, P. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[30] Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[31] Min, S., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. 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)*, pp. 5316–5330, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[32] Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., and Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[33] Mirza, M. J., Micorek, J., Possegger, H., and Bischof, H. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 14745–14755, 2022.

[34] Moon, T. K. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13 (6):47–60, 1996.

[35] Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[36] Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics.

[37] Oh, C., Hwang, H., Lee, H. Y., Lim, Y., Jung, G., Jung, J., Choi, H., and Song, K. Blackvip: Black-box visual prompting for robust transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 24224–24235, 2023.

[38] Pan, J., Gao, T., Chen, H., and Chen, D. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8298–8319, Toronto, Canada, July 2023. Association for Computational Linguistics.

[39] Pezeshkpour, P. and Hruschka, E. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.

[40] Pilehvar, M. T. and Camacho-Collados, J. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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)*, pp. 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[41] Platt, J. et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[42] Prasad, A., Hase, P., Zhou, X., and Bansal, M. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3845–3864, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[43] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pp. 8748–8763, 2021.

[44] Razeghi, Y., Logan IV, R. L., Gardner, M., and Singh, S. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 840–854, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[45] Roemmele, M., Bejan, C. A., and Gordon, A. S. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*, 2011.

[46] Rubin, O., Herzig, J., and Berant, J. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, Seattle, United States, July 2022. Association for Computational Linguistics.

[47] Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[48] Shin, S., Lee, S.-W., Ahn, H., Kim, S., Kim, H., Kim, B., Cho, K., Lee, G., Park, W., Ha, J.-W., and Sung, N. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5168–5186, Seattle, United States, July 2022. Association for Computational Linguistics.

[49] Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4222–4235, Online, November 2020. Association for Computational Linguistics.

[50] Si, C., Friedman, D., Joshi, N., Feng, S., Chen, D., and He, H. Measuring inductive biases of in-context learning with underspecified demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11289–11310, Toronto, Canada, July 2023. Association for Computational Linguistics.

[51] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[52] Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[53] Sorensen, T., Robinson, J., Rytting, C., Shaw, A., Rogers, K., Delorey, A., Khalil, M., Fulda, N., and Wingate, D. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 819–862, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[54] Wan, X., Sun, R., Dai, H., Arik, S., and Pfister, T. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 3493–3514, Toronto, Canada, July 2023. Association for Computational Linguistics.

[55] Wan, X., Sun, R., Nakhost, H., Dai, H., Eisenschlos, J. M., Arik, S. O., and Pfister, T. Universal self-adaptive prompting. *arXiv preprint arXiv:2305.14926*, 2023.

[56] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[57] Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3261–3275, 2019.

[58] Wang, D., Shelhamer, E., Liu, S., Olshausen, B. A., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[59] Wang, P., Li, L., Chen, L., Zhu, D., Lin, B., Cao, Y., Liu, Q., Liu, T., and Sui, Z. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023.

[60] Wei, J., Wei, J., Tay, Y., Tran, D., Webson, A., Lu, Y., Chen, X., Liu, H., Huang, D., Zhou, D., et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

[61] Williams, A., Nangia, N., and Bowman, S. 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)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[62] Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. An explanation of in-context learning as implicit bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[63] Xu, B., Wang, Q., Mao, Z., Lyu, Y., She, Q., and Zhang, Y. $k$nn prompting: Beyond-context learning with calibration-free nearest neighbor inference. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[64] Ye, S., Kim, D., Jang, J., Shin, J., and Seo, M. Guess the instruction! flipped learning makes language models stronger zero-shot learners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

[65] Yin, F., Vig, J., Laban, P., Joty, S., Xiong, C., and Wu, C.-S. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3063–3079, Toronto, Canada, July 2023. Association for Computational Linguistics.

[66] You, F., Li, J., and Zhao, Z. Test-time batch statistics calibration for covariate shift. *arXiv preprint arXiv:2110.04065*, 2021.

[67] Yuval, N. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[68] Zhang, T., Wang, X., Zhou, D., Schuurmans, D., and Gonzalez, J. E. TEMPERA: Test-time prompt editing via reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[69] Zhao, Z., Wallace, E., Feng, S., Klein, D., and Singh, S. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pp. 12697–12706, 2021.

[70] Zheng, C., Zhou, H., Meng, F., Zhou, J., and Huang, M. On large language models' selection bias in multi-choice questions. *arXiv preprint arXiv:2309.03882*, 2023.

[71] Zhou, H., Wan, X., Vulić, I., and Korhonen, A. Autopeft: Automatic configuration search for parameter-efficient fine-tuning. *arXiv preprint arXiv:2301.12132*, 2023.

[72] Zhou, H., Wan, X., Vulić, I., and Korhonen, A. Survival of the most influential prompts: Efficient black-box prompt search via clustering and pruning. *arXiv preprint arXiv:2310.12774*, 2023.

[73] Zou, Y., Zhang, Z., Li, C., Zhang, H., Pfister, T., and Huang, J. Learning instance-specific adaptation for cross-domain segmentation. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII*, pp. 459–476, 2022.

A Additional Related Work

Prompt Learning. Prompt learning is an efficient learning pipeline for LLM as an alternative to traditional full-model fine-tuning [27]. Soft prompting [24, 23, 28] enables fast adaptation of LLM by appending learnable continuous prompts in the embedding space while freezing the rest of weights. The recent development of parameter-efficient fine-tuning methods [18, 19, 71], which learn additional modules, may also be interpreted as a form of soft prompting [15]. However, these soft prompt learning methods inevitably require gradients and internal model access. On the other hand, hard prompting [49] is an appealing learning category for learning discrete prompts. Recent efforts have been devoted to black-box prompt search without accessing model gradients, and more interpretable prompts can be found by reinforcement learning [9, 68], gradient estimation [10], and other derivative-free search algorithms [42, 72].

Test-Time Adaptation. Test-time adaptation aims to mitigate the domain covariate shift using the test-time statistics. Wang et al. [58] propose TENT that minimizes the entropy by updating the affine parameters in the BN layer. Nado et al. [35] and Schneider et al. [47] introduce using test-time batch statistics for the standardization in BN and mixing it with source statistics to conquer covariate shift, respectively. Similarly, mixing the statistics with predefined hyperparameters [66, 22], interpolating source and target-domain statistics [25], or using a running average estimate [33, 20] have also been proposed to adapt the BN layer. Zou et al. [73] introduce strength parameters in adapting the standardization statistics in semantic segmentation tasks. We differentiate from test-time BN approaches by mitigating the bias in the novel context of LLM, and there is no source statistic similar to a BN layer in computer vision backbones.

B Experimental Setup

Evaluation Data. For natural language tasks, in contrast to previous works that only report on relatively simple single-sentence classification tasks [69, 12, 14], we conduct experiments on 13 more diverse and challenging classification tasks, including the standard GLUE [56] and SuperGLUE [57] datasets. Specifically, we consider commonsense reasoning: BoolQ [6], COPA [45]; word disambiguation: WiC [40]; sentiment classification: SST-2 [51]; paraphrasing: QQP, MRPC [11]; natural language inference and entailment: ANLI-R{1,2,3} [36], CB [8], RTE, QNLI (QA/NLI), MNLI [61]. For image classification tasks, we include SVHN [67], EuroSAT [16], and CLEVR [21].

Models. We conduct experiments mainly on the state-of-the-art PaLM 2 [2] for its variants with different sizes, PaLM 2-S, PaLM 2-M, and PaLM 2-L. PaLM 2 is trained using a mixture of objectives, and readers are referred to [2] for more details. For VLMs, we report the results on CLIP ViT-B/16 [43].

C Dataset Statistics

Table 3: Details of the dataset used for evaluation in the Table 2. |Test| denotes the number of test samples, where we consistently use the validation split as the test split because labels are not publicly available for some datasets.

Dataset	Objective	#sentences	#classes	|Test|
SST-2	Sentence Classification	1	2	872
MNLI	NLI	2	3	9815
QNLI	Question-Answering NLI	2	2	5463
MRPC	Paraphrasing	2	2	408
QQP	Paraphrasing	2	2	40430
BoolQ	Commonsense Reasoning	2	2	3270
CB	NLI	2	3	56
COPA	Commonsense Reasoning	3	2	100
RTE	NLI	2	2	277
WiC	Context Comprehension	3	2	638
ANLI-R1	NLI	2	3	1000
ANLI-R2	NLI	2	3	1000
ANLI-R3	NLI	2	3	1200

D Implementation Details

Contextual Calibration [69] (CC). We follow the original implementation of CC and take the mean of the log-probability over three content-free tokens as the test sample in the predefined template: 'N/A', '', '[MASK]'. It incurs 3 additional API costs from LLMs.

Domain-Context Calibration [12] (DC). We reproduce the DC baseline by using the same test set as the unlabeled text set to construct its bag-of-words. We then randomly sample tokens for an average length to form the content-free and in-domain input from the bag-of-words. This process is then repeated randomly for 20 times, and we take the mean of the log-probability following the original implementation. It incurs 20 additional API costs from LLMs.

Prototypical Calibration [14] (PC). For a fair comparison, we use the same test set as the unlabeled estimate set for PC. We follow the same hyper-parameters reported by PC with 100 maximum iterations for EM and 100 times random initialization for the whole learning process to stabilize its estimation. It is noteworthy that this number of repetition is costly and relatively slow, especially when the |Test| is large.

Batch Calibration (BC). In all reported experiments, we compute the correction log-probability term $\hat{\mathbf{p}}(y|C)$ once after all test samples are seen. In the n-shot ICL experiments reported in Table 2 and Fig. 8, the k-shot ICL is concatenating k random training sample per class. In the BCL experiment that uses labeled samples, we use $J \times 128$ randomly selected training samples as the labeled data. In the robustness study, we use 1 randomly sampled example as the context to study the performance of BC with respect to the ICL choices. We then conduct the ICL order experiment by re-ordering 4 randomly sampled ICL examples. The rest experiments are conducted on the standard 1-shot ICL setup.

E Additional Experiments

Adjustable Batch Calibration Layer (BCL). While BC is designed to be zero-shot and inference-only, it is also common that some *labeled* data are available. In this section, we describe a simple, adapted variant of BC that may further refine the calibration and mitigate any estimation errors from the unlabeled data, which we term *BCL*. Specifically, instead of deducting the bias term $\hat{\mathbf{p}}$ from the test data only, we introduce a single additional hyperparameter *strength* $\gamma \in \mathbb{R}$:

$$\mathbf{p}_{\text{BCL}}(y|x_i, C) = \mathbf{p}(y|x_i, C) - \gamma\hat{\mathbf{p}}(y|C), \quad (3)$$

where γ controls the strength of BC. To select the appropriate γ, we simply perform a grid search by uniformly sampling T different γ values in $[a, b]$ (we set $[a, b] := [-5, 5]$, but any reasonable range may be used). The strength γ is then learned by $\gamma^* = \arg\max_{\gamma \in [a,b]} R(\mathbf{p}_{\text{BC}}, \gamma)$, where $R(\cdot, \cdot)$ is the evaluation function (e.g., accuracy) on the set of *labeled* data, allowing the amount of calibration to be adjusted from evaluation metrics directly.

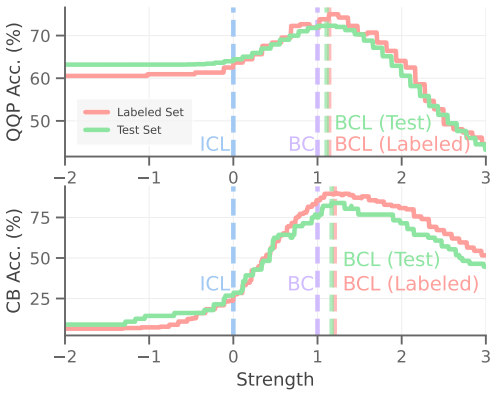

Figure 6: *BC benefits from labeled data:* The performance of an adaptable batch calibration layer (BCL) compared to the zero-shot BC with a changing strength. The `strength` γ at 0 and 1 represent the uncalibrated ICL and BC, respectively. We highlight the optimal strength learned from a labeled set by a red vertical line and the best test strength by a green line.

We give concrete examples in Fig. 6, which illustrates the effect of BCL where we plot the accuracy in QQP and CB tasks over a range of γ. We observe that $\gamma = 1$, which corresponds to BC without adjustment (purple line), leads to a strong but not optimal performance. By using the γ learned from the labeled data (a 128-shot randomly sampled set in this case), BCL estimates the contextual bias more precisely by leveraging the labeled data and achieves a performance that is very close to the optimal. We refer readers to Table 5 for more results.

Calibrating Vision-Language Models. Recently, vision-language models (VLM) [43], which simultaneously encode visual and textual information, have demonstrated strong zero-shot generalization capability by rewriting class labels. However, the sources of bias as LLMs have also been observed in prompting VLMs [1] but have not been adequately addressed. In this work, we propose to apply BC to Zero-Shot (ZS) CLIP [43] and mitigate the biases in its zero-shot classifications. We follow the same notation from Sec. 2, where the test image is now x, and the prompt template becomes the context, C. Similarly, we append the BC layer at the output of the ZS CLIP and calibrate for each class following Eq. 1 & 2.

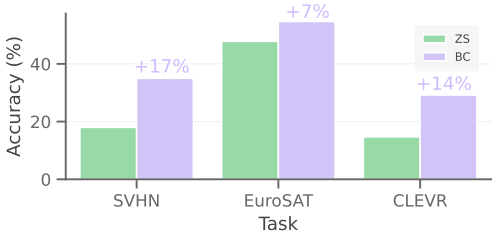

Figure 7: *BC improves zero-shot (ZS) image classification:* Accuracy (%) on image classification tasks with the zero-shot CLIP ViT-16/B. The BC implementation is zero-shot, and we apply BC together with the CLIP to demonstrate the effectiveness of BC in vision-language models. Refer to additional tasks in Appendix §E.

To handle the bias inherent in the prompt template designs in CLIP, we select three tasks in which the previous visual-prompting method shows significant improvement [37]. As shown in Fig. 7, BC significantly improves the zero-shot baseline by 12% on average. This observation further highlights the presence of contextual bias even within vision-language models, and BC can successfully restore the performance of VLM in image classification tasks, suggesting that BC may serve as a versatile and common technique for mitigating contextual biases across multiple modalities.

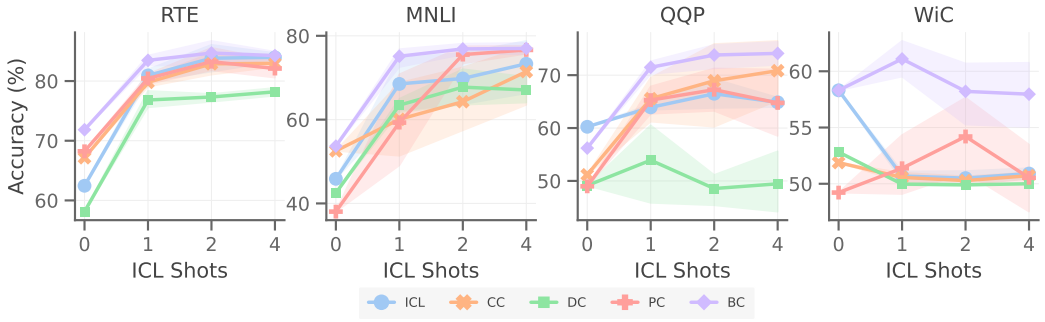

Figure 8: The ICL performance on various calibration techniques over the number of ICL shots on PaLM 2-S. Each shot indicates 1 example per class in the demonstration. Lines and shades denote the mean and standard deviation over 5 random seeds, respectively.

Table 4: Accuracy (%) on natural language classification tasks with 0-shot PaLM 2-S and 1-shot PaLM 2-M models in a single seed.

Model	PaLM 2-S 0-shot					PaLM 2-M 1-shot				
Method	ICL	CC	DC	PC	BC	ICL	CC	DC	PC	BC
SST-2	94.61	94.50	94.61	87.84	**95.18**	94.95	95.87	94.95	**96.22**	96.10
MNLI	45.87	52.54	42.50	38.04	**53.67**	45.50	54.43	56.26	43.81	**60.02**
QNLI	49.28	48.97	49.44	**50.28**	49.55	78.88	75.56	62.95	77.39	**78.91**
MRPC	69.12	61.76	**69.85**	**69.85**	64.95	57.11	**73.53**	68.87	69.85	65.93
QQP	**60.23**	51.16	49.12	48.98	56.20	66.18	**79.67**	74.32	70.27	75.13
BoolQ	86.51	**86.97**	76.88	55.41	84.04	87.37	88.53	87.28	**88.78**	87.31
CB	**85.71**	58.93	55.36	46.43	67.86	71.43	69.64	67.86	50.00	**80.36**
COPA	88.00	66.00	**90.00**	52.00	88.00	**97.00**	96.00	96.00	**97.00**	96.00
RTE	62.45	67.15	58.12	68.23	**71.84**	77.62	79.06	68.23	77.98	**80.51**
WiC	58.31	51.88	52.82	49.22	**58.30**	61.13	64.11	52.04	65.52	**68.03**
ANLI-R1	39.80	44.70	43.00	37.00	**50.00**	52.40	52.40	52.70	35.70	**54.00**
ANLI-R2	36.80	41.50	40.70	40.20	**45.10**	46.00	**50.70**	47.80	35.80	50.00
ANLI-R3	42.67	46.42	43.08	35.50	**48.50**	43.50	45.67	49.33	32.42	**50.50**
Avg.	63.03	59.42	58.88	52.23	**64.09**	67.62	71.17	67.58	64.67	**72.52**

Table 5: Accuracy (%) on natural language classification tasks with the zero-shot BC and the BCL. The experiments are evaluated with the same in-context example on 1-shot PaLM 2-S.

Method	SST-2	MNLI	QNLI	MRPC	QQP	BoolQ	CB	COPA	RTE	WiC	ANLI_{R1}	ANLI_{R2}	ANLI_{R3}	Avg.
BC	95.4	75.0	83.5	68.6	70.3	87.9	75.0	98.0	**84.1**	**63.3**	**59.8**	**51.1**	**53.3**	74.3
BCL	**96.3**	**75.0**	**83.5**	**74.3**	**72.3**	**88.8**	**83.9**	**99.0**	82.7	63.2	58.0	49.7	52.2	**75.3**

Table 6: Accuracy (%) on image classification tasks with the zero-shot CLIP ViT-16/B. We additionally report on UCF101 [52], FGVC Aircraft [30], and DTD [5].

Method	SVHN	EuroSAT	UCF	CLEVR	Aircraft	DTD	Avg.
ZS	18.0	47.8	**66.7**	14.7	**24.8**	**44.4**	36.7
ZS+BC	**35.0**	**54.7**	66.0	**29.2**	22.3	41.7	**41.5**

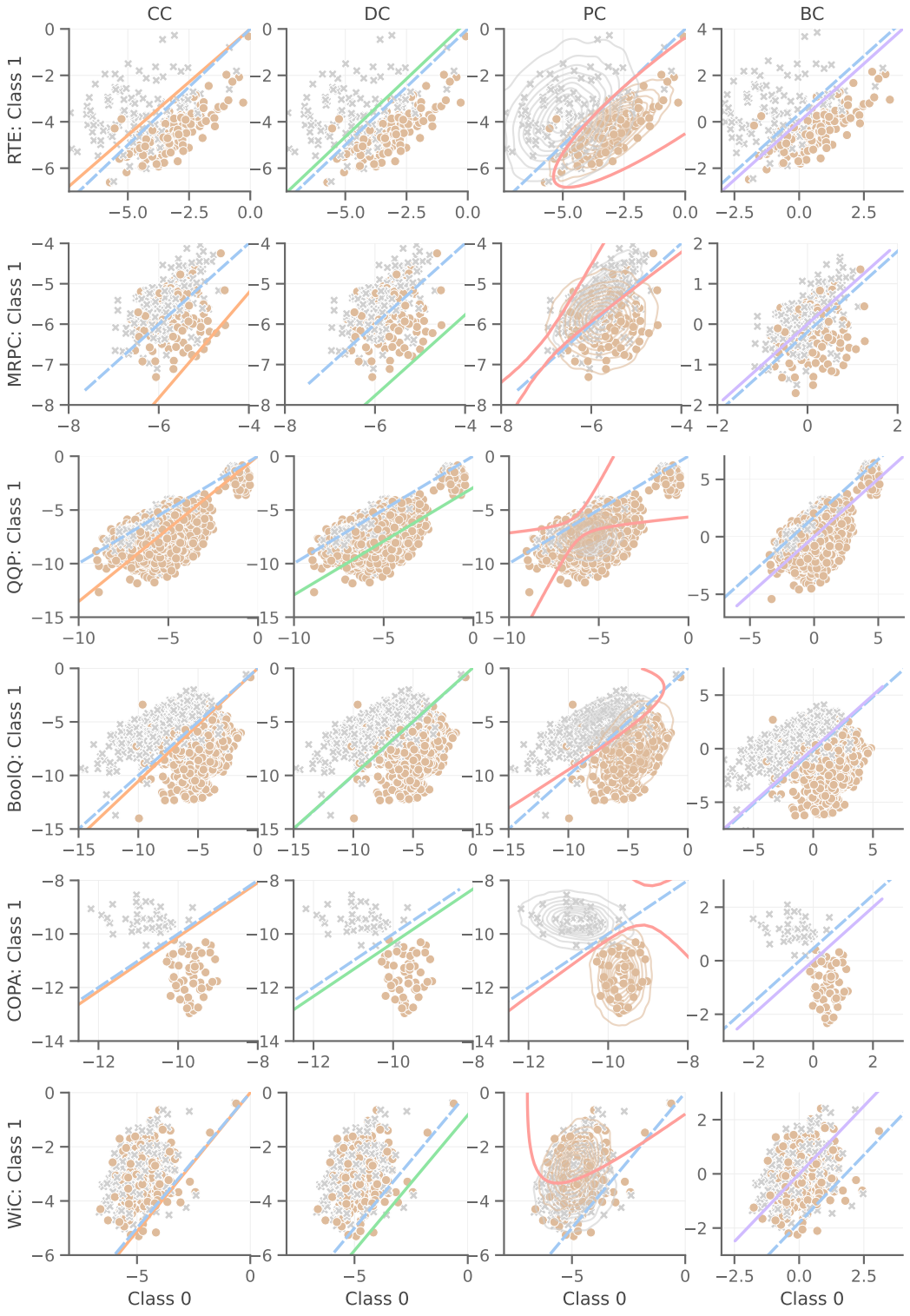

Figure 9: Visualization of the decision boundaries of uncalibrated ICL, and after applying existing calibration methods and the proposed BC. We list all binary classification tasks from the evaluation set.

F Prompt Templates

Table 7: Prompt templates for all k-shot ICL experiments. We follow the template styles from Han et al. [14] and Brown et al. [3].

Dataset	Template	Label Set
SST-2	Review: {sentence} Sentiment: {label}	negative / positive
MNLI CB ANLI	Premise: {premise} Hypothesis: {hypothesis} Answer: {label}	yes / maybe / no
QNLI	Question: {question} Sentence: {sentence} Label: {label}	yes / no
MRPC	Sentence 1: {sentence1} Sentence 2: {sentence2} Equivalence: {label}	no / yes
QQP	Question 1: {question1} Question 2: {question2} Duplicate: {label}	no / yes
BoolQ	{passage} Question: {question} Answer: {label}	no / yes
COPA	Premise: {premise} Choice1: {choice1} Choice2: {choice2} Answer: {label}	1 / 2
RTE	Premise: {sentence1} Hypothesis: {sentence2} Answer: {label}	yes / no
WiC	Sentence1: {sentence1} Sentence2: {sentence2} Word: {word} Answer: {label}	false / true

Table 8: Prompt templates for the robustness experiment conducted on RTE in Fig. 4.

ID	Template	Label Set
1	Premise: {sentence1} Hypothesis: {sentence2} Answer: {label}	yes / no
2	{sentence1} Hypothesis: {sentence2} Answer: {label}	
3	{sentence1} Question: {sentence2} Answer: {label}	
4	{sentence1} Question: {sentence2} {label}	
5	{sentence1} Question: {sentence2} yes or no? Answer: {label}	
6	Sentence 1: {sentence1} Sentence 2: {sentence2} Answer: {label}	
7	Premise: {sentence1} Hypothesis: {sentence2} Label: {label}	
8	Sentence 1: {sentence1} Sentence 2: {sentence2} Label: {label}	
9	Determine if the sentence 2 is true based on the Sentence 1 below Sentence 1: {sentence1} Sentence 2: {sentence2} Answer: {label}	
10	Determine if the sentence 2 is true or false based on the Sentence 1 below Sentence 1: {sentence1} Sentence 2: {sentence2} Answer: {label}	

Table 9: Verbalizer choices for the robustness experiment conducted on RTE in Fig. 4, where we include emoji pairs for ID 8, 9, 10.

ID	Label Set	Template
1	yes / no	Premise: {sentence1} Hypothesis: {sentence2} Answer: {label}
2	true / false	
3	correct / incorrect	
4	positive / negative	
5	good / bad	
6	great / terrible	
7	it was true / it was false	
8	:thumbs_up / :thumbs_down	
9	:man_gesturing_ok / :man_gesturing_no	
10	:check_mark / :cross_mark	

Table 10: Prompt templates for the 0-shot experiments.

Dataset	Template	Label Set
SST-2	Review: {sentence} Sentiment: {label}	negative / positive
MNLI CB ANLI	{premise} Question: {hypothesis} yes, no, or maybe? Answer: {label}	yes / maybe / no
QNLI	{question} Question: {sentence} yes or no? Answer: {label}	yes / no
MRPC	Sentence 1: {sentence1} Sentence 2: {sentence2} Equivalence: {label}	no / yes
QQP	Question 1: {question1} Question 2: {question2} Duplicate: {label}	no / yes
BoolQ	{passage} Question: {question} Answer: {label}	no / yes
COPA	Premise: {premise} Choice1: {choice1} Choice2: {choice2} Answer: {label}	1 / 2
RTE	{sentence1} Question: {sentence2} yes or no? Answer: {label}	yes / no
WiC	{sentence1} {sentence2} Question: Is the word '{word}' used in the same way in the two sentences above? Answer: {label}	no / yes