Beyond Performance: Quantifying and Mitigating Label Bias in LLMs

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated impressive adaptability to diverse tasks, by relying only on context prompts containing instructions, or minimal input-output examples. However, recent work revealed they also exhibit *label bias*—an undesirable preference toward predicting certain answers over others. Still, detecting and measuring this bias reliably and at scale has remained relatively unexplored. In this study, we evaluate different approaches to quantifying *label bias* in a model's predictions, conducting a comprehensive investigation across 279 classification tasks and ten LLMs. Our investigation reveals substantial label bias in models both before and after debiasing attempts, as well as highlights the importance of outcomes-based evaluation metrics, which were not previously used in this regard. We further propose a novel label bias calibration method tailored for few-shot prompting, which outperforms recent calibration approaches for both improving performance and mitigating label bias. Nevertheless, our results emphasize that label bias in the predictions of LLMs remains a barrier to their reliability.[1]

1 Introduction

Large language models (LLMs) have shown remarkable abilities in adapting to new tasks when conditioned on a context prompt, containing task solving instructions (Wei et al., 2022) or few examples of input-output pairs (Brown et al., 2020). Still, recent work has shown that predictions of LLMs exhibit *label bias*—a strong, undesirable preference towards predicting certain answers over others (Zhao et al., 2021; Chen et al., 2022; Fei et al., 2023, see Fig. 1). Such preferences were shown to be affected by the choice and order of in-context demonstrations (Liu et al., 2022; Lu et al., 2022), the model's pretraining data (Dong et al., 2022), or textual features of the task data (Fei et al., 2023). Consequently, several approaches were proposed to address this problem, mostly by calibrating the model's output probabilities to compensate for this bias (Zhao et al., 2021; Fei et al., 2023).

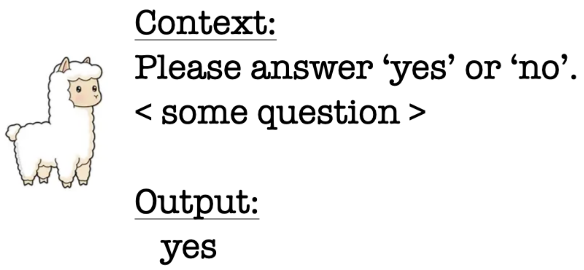


Figure 1: LLMs exhibit *label bias*—a tendency to output a given label regardless of the context (in this example, 'yes' over 'no'). In this work we evaluate LLM label bias across ten LLMs and 279 classification tasks, showing label bias is a major problem in LLMs.

Despite these efforts, label bias evaluation relies on *performance* metrics such as accuracy, rather than metrics designed to directly measure the *bias*. In doing so, we might inadvertently overlook crucial aspects of model behavior. Indeed, although a given method could effectively improve performance, substantial bias might still persist in the model's predictions—deeming the method insufficient and the model unreliable. Alternatively, performance could remain relatively unchanged, but with the bias mostly removed.

In this work, we take a step towards a more comprehensive understanding of the extent of label bias in LLMs and the effects of mitigation approaches. Using metrics to quantify label bias in model predictions, which we derive from previous work on fairness and label bias estimation, we evaluate ten LLMs on 279 diverse classification tasks from SUPER-NATURALINSTRUCTIONS (Wang et al., 2022). We examine both performance and bias along axes such as scale and number of in-context

[1] We will release our code upon publication.

demonstrations. We also evaluate the impact of label bias mitigation methods, such as calibration and few-shot LoRA fine-tuning (Hu et al., 2022).

Our investigation reveals substantial label bias in the predictions of LLMs across all evaluated settings, indicating that raw LLM output scores often represent simple, heuristic solutions. While increasing model size, providing in-context demonstrations, and instruction-tuning all contribute to reducing bias, ample bias persists, even after applying mitigation methods. Surprisingly, these results also hold for tasks where the labels are all semantically equivalent (e.g., in multi-choice question answering). Further, although the examined calibration methods can reduce bias and improve performance, we also find cases where they negatively impact both bias and overall performance.

Motivated by these findings, we propose a novel calibration method for few-shot prompting that accurately estimates a model's label bias using only its predictions on the prompt's in-context demonstrations. Compared to existing LLM calibration methods, our method improves performance while also removing considerably more bias.

Our findings highlight the necessity of considering and measuring biases in the predictions of LLMs whenever benchmarking their performance. Furthermore, adapting models to their tasks through more accurate and effective estimation of biases, as demonstrated by our proposed method for calibrating few-shot prompting, offers a promising avenue for improving the reliability of LLMs and their applications.

2 LLM Label Bias

Our objective is to broaden the understanding of label bias in LLMs and the effectiveness of mitigation strategies, focusing on classification tasks. In this section, we define metrics designed to quantify bias in model predictions, aiming to provide a nuanced examination of label bias that extends beyond traditional performance metrics. We describe the setting of label bias in in-context learning (§2.1), and then review approaches to evaluating it and define the metrics we use in this work (§2.2).

2.1 Label Bias

When employing LLMs for classification tasks through prompting, the model is given a test example x, preceded by a context C. This context can contain a (potentially empty) set of examples of the task's input-output mapping $[(x^1, y^1), \dots, (x^k, y^k)]$, henceforth *demonstrations*, and may also include task instructions. To determine the model's prediction from a set of answer choices Y, the likelihood it assigns to each continuation $y \in Y$ is computed, and the highest probability option is taken as the model prediction:

$$\arg\max_{y \in Y} p(y \mid x, C)$$

These output probabilities often exhibit *label bias*, where the model tends to assign higher probability to certain answers regardless of the input test example x (Fig. 1). Multiple factors were posited to influence this bias, including the choice of verbalizers Y, the choice and order of in-context examples in C, and the overall textual features of task input x (Zhao et al., 2021; Fei et al., 2023).

2.2 Evaluation Measures

Most analyses of LLM label bias rely on indirect assessments, based on inspecting improvements in overall performance gained after applying techniques to mitigate it (Fei et al., 2023; Holtzman et al., 2021; Zhao et al., 2021). However, these do not indicate the extent of bias originally present, or that remains after mitigation. We next examine approaches to measure this bias more directly, and define the metrics we use in this work. Importantly, we focus on label bias measures that could be used effectively both before and after applying mitigation techniques such as calibration.

Drawing from previous research on fairness and bias in machine learning, we observe that there are two distinct yet related aspects in which label bias can be measured in LLM predictions: through the probabilities assigned by the model to different answers; and through the model's final predictions compared to the gold labels (Mehrabi et al., 2021).

Probabilistic approach To assess the first, probabilistic aspect, previous work used qualitative assessments to visualize model output distributions on selected datasets (Zhao et al., 2021; Han et al., 2023). However, these cannot be used to rigorously evaluate model behavior on larger scales. Recently, Fei et al. (2023) proposed to measure the model's label bias by comparing its mean output probabilities $\hat{p}_{cf}$ on synthetic and "content-free" task inputs $\hat{X}_{cf}$, built by concatenating random words from the task's test data, against the model's output probabilities $\hat{p}_{rand}$ on inputs consisting of random vocabulary words $\hat{X}_{rand}$. These output distributions are computed over the set of answer choices Y, by taking the model's average output probabilities for each label $y \in Y$ across the two sets of inputs:

$$\hat{p}_*(y) = \frac{1}{|\hat{X}_*|} \sum_{x \in \hat{X}_*} p(y \mid x, C)$$

The model's bias is then defined to be the total variation distance d_{TV} between both distributions:

$$d_{TV}(\hat{p}_{\text{cf}}, \hat{p}_{\text{rand}}) = \frac{1}{2} \sum_{y \in Y} |\, \hat{p}_{\text{cf}}(y) - \hat{p}_{\text{rand}}(y) \,|$$

Importantly, since Fei et al. (2023) also use the model's predictions on $\hat{X}_{cf}$ for calibration, this metric cannot be used to quantify the label bias remaining after calibration.

In this work, we simplify the computation of this metric and adapt it to be used after calibration. First, we hold-out a set of inputs to be used exclusively for measuring bias. Second, when estimating the model's average output probabilities, instead of using randomly concatenated words, we use in-distribution examples extracted from the test set, $\hat{X}_{\text{i.d.}} = ((x_1, y_1), \ldots, (x_m, y_m))$. This setup allows to account for label imbalance in the data used for bias estimation $\hat{X}_{\text{i.d.}}$, as the instances in the test set are all labeled. To do so, we first estimate the model's output distribution individually on each subset of examples with gold label $\ell \in Y$, $\hat{X}^{\ell}_{\text{i.d.}} = \{(x, y) \in \hat{X}_{\text{i.d.}} \mid y = \ell\}$, by computing:

$$\hat{p}^{\ell}_{\text{i.d.}}(y) = \frac{1}{|\hat{X}^{\ell}_{\text{i.d.}}|} \sum_{x \in \hat{X}^{\ell}_{\text{i.d.}}} p(y \mid x, C)$$

and then set $\hat{p}_{\text{i.d.}}$ to be the average of these estimates.[2] Instead of $\hat{p}_{rand}$, we use the uniform distribution over all answer choices $(\frac{1}{|Y|}, \ldots, \frac{1}{|Y|})$, which recent mitigation approaches consider as the "ideal", unbiased output distribution (Zhao et al., 2021). Finally, we define the model's **bias score** as the total variation distance between these two distributions:

$$BiasScore = \frac{1}{2} \sum_{y \in Y} \left| \, \hat{p}_{\text{i.d.}}(y) - \frac{1}{|Y|} \, \right|$$

[2]In cases where examples for an infrequent label $\ell \in Y$ are not found in $\hat{X}_{i.d.}$, we do not take it into account when computing $\hat{p}_{\text{i.d.}}$.

Outcome-based approach When considering the effects of label bias on model predictions, strong label bias will likely result in disparities in task performance on instances of different classes. However, metrics to assess such disparities were not used in previous analyses of label bias.

We propose to use the **Relative Standard Deviation of class-wise accuracy** (*RSD*; Croce et al. 2021; Benz et al. 2021), a metric used for studying fairness in classification. *RSD* is defined as the standard deviation of the model's accuracy per class ($acc_1, \ldots, acc_{|Y|}$), divided by its mean accuracy acc on the entire evaluation data:[3]

$$RSD = \frac{\sqrt{\frac{1}{|Y|} \sum_{i=1}^{|Y|} (acc_i - acc)^2}}{acc}$$

Intuitively, *RSD* is low when model performance is similar on all classes, and high when it performs well on some classes but poorly on others.

Discussion We note that each evaluation approach could detect biases that the other does not. For example, a slight bias in the model's average output probabilities (e.g., 55% vs. 45%) could render dramatic bias in actual outcomes if the model *always* assigns higher probability to some label. Conversely, when the output probabilities are biased *on average* but the model's class-wise performance is balanced, this *hidden* bias could result in actual performance disparities in more difficult cases. We therefore report both metrics in this work.

3 Experimental Setting

3.1 Datasets

We evaluate models on 279 diverse tasks from the SUPER-NATURALINSTRUCTIONS benchmark (Wang et al., 2022). We select all available classification and multi-choice question answering tasks where the output space is a set of predefined labels, such as "A/B/C" or "positive/negative". We sample 1,000 examples for evaluation for all tasks with larger data sizes, and additionally sample 32 held-out examples for computing the bias score metric (§2.2), and 64 more examples to be used as a pool of instances for choosing in-context demonstrations and LoRA fine-tuning examples. We only include tasks with at least 300 evaluation examples in our experiments.

[3]The goal of this normalization is to enhance the metric's interpretability across tasks of varying difficulty.

3.2 Models and Evaluation Setup

We experiment with models of different sizes from three LLMs families: LlaMA-2 7B and 13B (Touvron et al., 2023), Mistral 7B (Jiang et al., 2023a), and Falcon 7B and 40B (Penedo et al., 2023). We use both the base and instruction fine-tuned versions of each model. We evaluate models using context prompts with $k \in \{0, 2, 4, 8, 16\}$ demonstrations, and average the results across 3 different sets of demonstrations for each k. To control the evaluation budget, we run the more expensive LoRA and Falcon 40B experiments with $k \in \{0, 8, 16\}$ averaged across 2 sets of demonstrations.

We use the task instructions and prompt template defined in SUPER-NATURALINSTRUCTIONS. For tasks where the answer choices $y \in Y$ have unequal token lengths, we use length-normalized log-likelihood when computing the model's output probabilities (Holtzman et al., 2021). For further implementation details, see App. A.1.

Data contamination During their instruction tuning, Llama-2 chat models were initially fine-tuned on the *Flan* data collection (Chung et al., 2022; Longpre et al., 2023), approximately 20% of which is comprised of examples from the SUPER-NATURALINSTRUCTIONS benchmark. Therefore, our evaluation of the Llama-2 instruction-tuned models is likely effected by data contamination (Magar and Schwartz, 2022). Still, our results show both models exhibit extensive label bias, possibly due to later fine-tuning on other data. As it is unclear from the implementation details of Touvron et al. (2023) which examples in SUPER-NATURALINSTRUCTIONS were included in training, we do not take extra steps in attempt to reduce possible overlap and contamination.

3.3 Bias Mitigation Techniques

We evaluate the effects of three label bias mitigation methods: two calibration methods designed to correct a model's label bias by adjusting its output scores; and few-shot LoRA fine-tuning (Hu et al., 2022), which adapts the model to the task and its label distribution. We describe each method below.

Contextual calibration (CC) Zhao et al. (2021) proposed to use calibration in order to remove the label bias arising from the context prompt C and the model's pretraining. Inspired by confidence calibration methods (Guo et al., 2017), they define a matrix W that is applied to the model's original output probabilities p to obtain calibrated, debiased probabilities $q = \text{softmax}(Wp)$. To determine the calibration parameters W, they first compute the model's average predicted probabilities $\hat{p}$ on a small set of "placeholder", content-free input strings such as "[MASK]", which replace the task input that follows C.[4] They then set $W = \text{diag}(\hat{p})^{-1}$, so that the class probabilities for the average content-free input would be uniform, aiming to remove the model's underlying bias.

Domain-context calibration (DC) Following the CC method, Fei et al. (2023) proposed to capture the label bias resulting from the word distribution of the task dataset when estimating $\hat{p}$. They constructed in-domain yet content-free inputs by sampling and concatenating L random words from the test set, where L is the average instance input length in the data. They repeat this process $M = 20$ times, and set $\hat{p}$ to be the average output probabilities over all M examples. Given a test example with original output probabilities p, they then use the calibrated probabilities $q = \text{softmax}(p/\hat{p})$ for prediction.

Few-shot fine-tuning Finally, we also experiment with few-shot, parameter-efficient fine-tuning as an effective approach for adapting LLMs to a given task's label distribution, thus potentially mitigating label bias. We fine-tune task-specific models for each context prompt using Low-Rank Adapation (LoRA; Hu et al., 2022), training adapters on 16 held-out training examples for 5 epochs. Importantly, we use the same context C during both fine-tuning and evaluation. Due to computational constraints, we only run this method on Llama-2 7B and Mistral 7B. See App. A.3 for additional details.

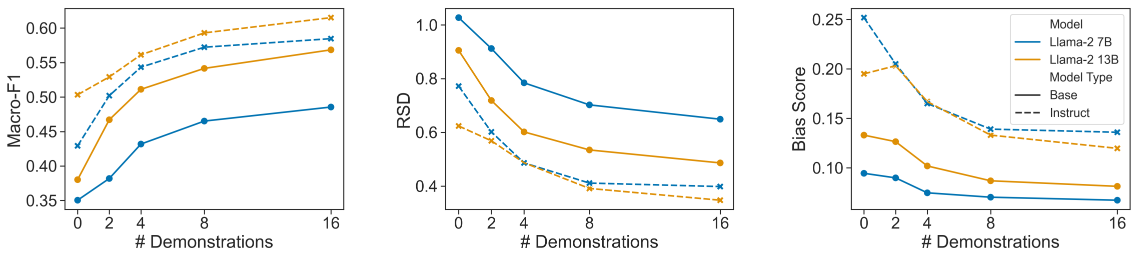
4 Quantifying Label Bias in LLMs

4.1 LLMs are Label-Biased

We begin by examining the performance and label bias of models with and without instruction-tuning. We report averaged results across all tasks for Llama-2 models in Fig. 2. Results for other models show similar trends, and are found in App. B.1.

We first verify that, as expected, model performance (Fig. 2a) substantially improves with scale, with instruction tuning and with the number of demonstrations. We then consider the two bias

[4] As in the original implementation, we use "N/A", "[MASK]" and the empty string.



(a) Performance (Macro-F1)

(b) Label bias (*RSD*)

(c) Label bias (*BiasScore*)

Figure 2: Performance (higher is better) and label bias metrics (lower is better) for Llama-2 pretrained and instruction-tuned models (7B/13B). Both performance and *RSD* improve with scale, instruction tuning, and number of demonstrations. In contrast, *BiasScore* does not improve with scaling, and is worse after instruction-tuning.

metrics—*RSD* (Fig. 2b) and *BiasScore* (Fig. 2c). We observe that label bias is substantial across most evaluation settings: All models obtain *RSD* of around 0.40 at their best evaluated setting, and reach values close to 1 at their worst. This implies a widespread disparity in model performance across classes in many of the evaluated tasks, indicating that, for most tasks, they succeed only on instances of certain classes, while consistently failing on instances from others.

Conversely, while *BiasScore* is relatively high for some, most models obtain values around 0.1. This indicates that the averaged output probabilities are relatively close to uniform. Taken together with *RSD*, this hints that LLM label bias is often not the result of a highly skewed output distribution that automatically assigns high probability to preferred classes. Rather, it stems from close-to-uniform probability in cases of uncertainty, failing to capture the correct answer for less favored classes.

4.2 Differences between the Bias Measures

We further note that, interestingly, both bias metrics show divergent trends. Although *RSD* values, much like model performance, sharply improve after instruction-tuning, the resulting models' *BiasScore* is often higher than their vanilla counterparts. Similarly, while *RSD* improves substantially with scaling, *BiasScore* of smaller models are lower.[5]

We note that higher performance together with lower *RSD* means that the model's performance has improved across most classes. In contrast, higher *BiasScore* implies that its average predicted probabilities grew farther than uniform. As a result, the discrepancy between the metrics indicates that the scaled-up and instruction-tuned models are making more confident predictions on some classes, but not on others. This could either mean more confident correct predictions on the preferred classes, or more confidently wrong predictions on others (or both). Altogether, this suggests that more subtle forms of bias persist after instruction-tuning or scaling up (Tal et al., 2022).

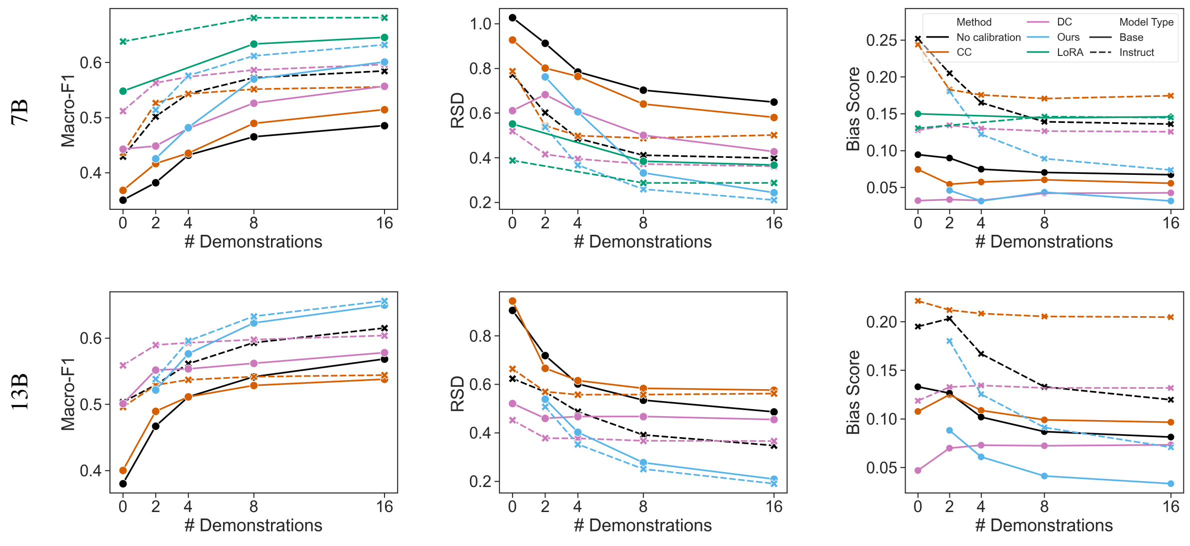
Overall, we find the two metrics to be complimentary due to their measuring of different aspects of label bias. We hence use both in further experiments to provide a more comprehensive understanding of such bias in model predictions.

4.3 Label Bias Persists after Mitigation

We have seen that LLMs demonstrate extensive label bias across different models, scales and tasks (§4.1). We next examine techniques aimed at mitigating such bias, and assess the extent of label bias remaining after their application. We report our results for Llama-2 models in Fig. 3. We observe similar trends for other models, and report their results in App. B.2.

We first consider the effect of bias mitigation on model performance (Fig. 3a) using the three methods described in §3.3: contextual calibration (CC), domain-context calibration (DC), and few-shot fine-tuning with LoRA. Compared to standard prompting (**black** lines), we find that applying CC (**orange**) provides little to no gains. Moreover, it can even undermine model performance, especially for instruction-tuned models, as previously observed by Fei et al. (2023). In contrast, DC (**purple**) can provide substantial performance gains, specifically when using few or no in-context demonstrations, where baseline performance is relatively low. However, when calibrating instruction-

[5]Still, we note that *BiasScore* is not inversely correlated with model performance, as some models with high performance like Mistral-7B also have relatively low *BiasScore* (App. B.2).



(a) *Macro-F1* (b) *RSD* (c) *Bias Score*

Figure 3: The effect of label bias mitigation methods on performance and bias for Llama-2 models. CC improves neither performance nor bias; DC and LoRA fine-tuning improve both; our *Leave-One-Out Calibration* (LOOC) method leads to the best performance among the calibration methods, and the overall lowest *RSD* values with 8 or 16 demonstrations.

tuned models prompted with higher number of demonstrations, we find that DC mostly fails to improve performance. Finally, LoRA considerably improves performance in all cases (**green** in Fig. 3, upper row), vastly outperforming both CC and DC.

We next turn to measure label bias (Fig. 3b and 3c). Notably, unlike for scale and instruction-tuning, here *BiasScore* also roughly mirrors the changes in model performance due to calibration, though this is not the case for LoRA.[6] In consequence, both calibration methods fail to mitigate label bias in the cases mentioned above. As for LoRA, the best *RSD* results are still around 0.3, and *BiasScore* noticeably increases after fine-tuning, indicating that more subtle bias persists.

Overall, our results indicate that existing bias calibration approaches are insufficient for diminishing label bias in essential cases, particularly for instruction-tuned models. Further, while LoRA fine-tuning is effective in both improving performance and mitigating certain aspects of bias (though not others), it is also substantially more computationally expensive than calibration.

5 Mitigating Label Bias by Calibrating on Demonstrations

Motivated the failures of existing calibration approaches on instruction-tuned models (§4.3), we aim to develop an effective calibration method for such scenarios. We hypothesize a possible cause for the observed failures is that the inputs used for calibrating label bias in these methods are very distinct from the more curated, high-quality inputs models observe during instruction-tuning (Touvron et al., 2023).[7] Similarly, although pretraining corpora are known to contain lower quality data (Marion et al., 2023), the unusual qualities of inputs used in these methods could also hinder potential further gains on pretrained models.

Seeking to use more naturally-occurring inputs, yet aiming to avoid reliance on additional test set examples, we propose to calibrate models using the in-context demonstrations used in few-shot prompting. However, since these examples appear alongside their labels in the context, naively obtaining the model's output probabilities for calibration would result in unreliable bias estimates. We next introduce a simple method to alleviate this concern.

[6] In other words, changes in *BiasScore* are generally sufficient to determine changes in performance.

[7] Specifically, nonsensical task inputs made up of random words as in DC, or placeholder-like strings as in CC, are less likely to be observed during instruction tuning.

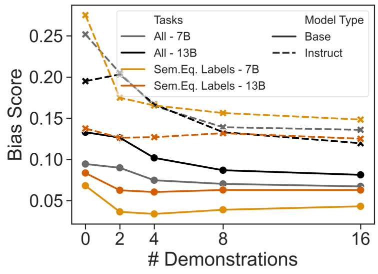
Leave-One-Out Calibration (LOOC) Our goal is to estimate the model's average output probabilities $\hat{p}$ at test-time by using the k demonstrations $[(x^1, y^1), \ldots, (x^k, y^k)]$ provided in the context C, and then use this estimate for calibration. Drawing from leave-one-out cross-validation, when evaluating the model on the i-th demonstration's input x^i, we prompt it with an edited context C_i comprised of the original context C after removing the current demonstration (x^i, y^i).[8] We thus obtain model output probabilities $p^1, \ldots, p^k$, each prompted with $k-1$ labeled demonstrations.

To reliably estimate $\hat{p}$, we further need to account for the demonstrations' labels y^i: for imbalanced choices of demonstrations (e.g., for tasks with imbalanced classes), using the average of p^i's could lead to an underestimation of the probability assigned to infrequent labels. We therefore compute the average output probabilities $\hat{p}$ by taking into account the labels y^i, as we do for computing *BiasScore* (§2.2). We first average p^i's associated with the same label, and then set $\hat{p}$ as the simple average of these intra-label averages. Finally, we use the estimate $\hat{p}$ to compute calibration parameters and score new examples using the same methodology as Zhao et al., 2021 (§3.3). We refer to our method as Leave-One-Out Calibration (LOOC).

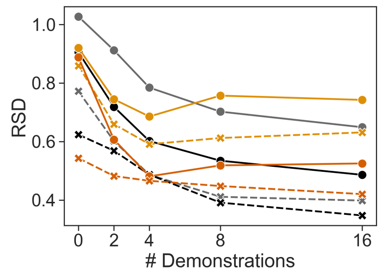
Results We use LOOC to calibrate models in the same setup used for other bias mitigation approaches (§3). We report our results for Llama-2 models in Fig. 3 (**cyan** lines), finding similar trends in other models (App. B.2). Comparing our method to other calibration approaches, we find LOOC surpasses both CC and DC by a wide margin in performance and bias metrics for context prompts with $k = 8, 16$. Importantly, using LOOC to calibrate instruction-tuned models in this setting dramatically improves upon the uncalibrated model, whereas other methods fail to achieve meaningful gains (§4.3). Further, LOOC nearly closes the gap with LoRA-level performance, as well as improves upon it in both bias metrics, while requiring substantially less computational resources.

As LOOC relies on the in-context demonstrations for bias estimation, k needs to be sufficiently large for calibration to succeed. Surprisingly, we find that with as few as $k = 4$ demonstrations, our method is often comparable to the next best calibration method on all metrics. Finally, we note that although our method can substantially reduce label bias compared to other approaches, the remaining *RSD* is non-negligible and indicates that model performance could still be biased on some tasks.

[8] We leave all other demonstrations in their original order.



(a) *BiasScore*



(b) *RSD*

Figure 4: Label bias metrics for Llama-2 models (7B/13B), when evaluated on all tasks in our evaluation suite (*All*) vs. a subset of tasks with semantically equivalent labels (*Sem.Eq. Labels*). LLMs exhibit label bias even on tasks with semantically equivalent labels, such as multi-choice question answering.

6 Label Bias for Semantically Equivalent Labels

The output space for classification tasks often consists of labels with strong semantic meanings (e.g., "Positive" vs. "Negative"). Recent work has indicated that when such labels are used for classification tasks, the model's decision could be affected by biases from their pretraining (Zhao et al., 2021), and that replacing the verbalizers used to denote labels often impacts model performance (Wei et al., 2023; Cui et al., 2022; Fei et al., 2023).

We next examine whether models exhibit less label bias when the task's labels are semantically equivalent and interchangeable,[9] and are thus less likely to be affected by model biases from pretraining. Most of the tasks in our evaluation suite (§3.1) have labels with meaningful and often opposed semantic meanings. We therefore extract a subset of

[9] E.g., the answers 1 and 2 represent other concepts introduced in the prompt, and their order could essentially be changed if we modify the prompt accordingly.

tasks with semantically equivalent labels. We extract all multi-choice QA tasks—with label spaces such as “A/B/C/D” or “1/2/3”—and all sentence completion tasks, where the model is tasked with choosing the more logical continuation for an input sentence between two provided options, usually labeled A and B. This results in 18 tasks with semantically equivalent labels.

We compare each model’s label bias on this subset of tasks and the entire evaluation suite for Llama2 models in Fig. 4, with results for other models largely following similar trends. We find that models exhibit extensive bias in terms of *RSD* on tasks with semantically equivalent labels, and in similar magnitude to their overall *RSD* across all tasks. We note that for pretrained models, *BiasScore* decreases on semantically equivalent tasks, but *RSD* remains high. Overall, this indicates that LLMs exhibit considerable label bias even when all labels are semantically equivalent in the context of their tasks.

7 Related Work

Biases in LLM predictions Recent work has revealed various biases in the predictions of LLMs. Wang et al. (2023a) showed that models are biased towards certain positions when presented with several texts for evaluation and ranking. Pezeshkpour and Hruschka (2023) showed that models are biased towards choosing answers in specific positions when tasked with multi-choice QA, while Zheng et al. (2023) propose a method to mitigate this debias. Si et al. (2023) exposed inductive biases of models during in-context learning. Lu et al. (2022) showed that the order of demonstrations in the context can greatly effect model predictions. Complimentary to these works, we focus on studying label bias in LLMs (Fei et al., 2023; Zhao et al., 2021) and seek to improve its evaluation.

Calibrating Bias in LLMs Recent work proposed methods to calibrate bias in LLMs, among which Zhao et al. (2021) and Fei et al. (2023) are included in our studies. Han et al. (2023) proposed to calibrate models by fitting a Gaussian mixture distribution to the model’s output probabilities, using this mixture for inference on new examples. However, they require several hundred labeled examples for calibration. Concurrently to our work, Jiang et al. (2023b) proposed to generate inputs for model calibration by prompting models with the context prompt, and Zhou et al. (2023) proposed to calibrate models by using their output probabilities on the entire test set. While the motivation for these methods is similar to our proposed calibration method, i.e., calibrating models by using inputs that are more naturally-occurring, our method does not require access to the test set, or additional computation to obtain inputs for calibration. Importantly, unlike previous work on bias calibration, our main focus is the evaluation of label bias and of bias mitigation methods in LLMs.

8 Conclusion

The label bias of LLMs substantially hinders their reliability. We considered different approaches to quantifying this bias. Through extensive experiments with ten LLMs and across 279 classification tasks, we found that substantial amounts of label bias exist in LLMs. Moreover, we showed that this bias persists even as LLMs increase in scale, are instruction-tuned, are provided in-context demonstrations, and even when they are calibrated against such bias. We proposed a novel calibration method, which outperforms existing calibration approaches, and reduces label bias dramatically. Our results highlight the need to both better estimate and mitigate LLM label bias.

Limitations

Model sizes Although we experiment with models of several sizes, the models we use are all in the 7B-40B range. We chose not to include relatively small models as these often exhibit poor performance in prompt-based settings. While recent efforts have released better and more efficient models, we leave those for future work. We chose not to experiment with very large LLMs such as Llama 70B due to limitations in computational resources, and as many of them (e.g., GPT-4) are closed (Rogers et al., 2023). It is therefore unclear whether our findings apply to such models.

Prompt format Our evaluations are performed on a large and diverse set of tasks extracted from SUPER-NATURALINSTRUCTIONS. Still, all tasks contain similar prefixes before introducing instructions, demonstrations and task inputs. Furthermore, each task only has one human-written instruction. We leave experimentation with more varied formats and examination of bias across different instruction phrasings to future work.

Evaluating multilingual tasks To build our evaluation suite, we extracted tasks from SUPER-NATURALINSTRUCTIONS, focusing only on English tasks. We leave analysis on label bias for multilingual tasks to future work.

References

Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. 2021. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. In *NeurIPS 2020 Workshop on Pre-registration in Machine Learning*, pages 325–342. PMLR.

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. In *Advances in neural information processing systems*.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2022. On the relation between sensitivity and accuracy in in-context learning. arxiv:2209.07661.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. arXiv:2210.11416.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. 2021. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. arXiv:2203.09770.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5937–5947, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031, Toronto, Canada. Association for Computational Linguistics.

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

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. 2021. 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*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. arXiv:2310.06825.

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

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. 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*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. arXiv:2301.13688.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. 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)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association* *for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When less is more: Investigating data pruning for pretraining llms at scale. arXiv:2309.04564.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. arxiv:2306.01116.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. arXiv:2308.11483.

Anna Rogers, Niranjan Balasubramanian, Leon Derczynski, Jesse Dodge, Alexander Koller, Sasha Luccioni, Maarten Sap, Roy Schwartz, Noah A Smith, and Emma Strubell. 2023. Closed ai models make bad baselines.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. 2023. 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)*, pages 11289–11310, Toronto, Canada. Association for Computational Linguistics.

Yarden Tal, Inbal Magar, and Roy Schwartz. 2022. Fewer errors, but more stereotypes? the effect of model size on gender bias. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 112–120, Seattle, Washington. Association for Computational Linguistics.

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

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators. arXiv:2305.17926.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023b. How far can camels go? exploring the state of instruction tuning on open resources. *arXiv preprint arXiv:2306.04751*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. arXiv:2303.03846.

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.

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.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. On large language models' selection bias in multi-choice questions. arXiv:2309.03882.

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

A Experimental Setting

A.1 Additional Implementation Details

Our implementation and pretrained model checkpoints use the Huggingface Transformers library (Wolf et al., 2020). When running inference, we load all models using bf16, except for Falcon-40B, which we load using 8bit inference, following Wang et al. (2023b). We run all experiments on Quadro RTX 6000 (24GB) and RTX A6000 (48GB) GPUs, except for Falcon-40B experiments, which we run on A100 GPUs. Average inference run-times on our entire evaluation suite is 18 hours for 7B models, 24 hours for 13B models, and 24 hours for 40B models. Running LoRA fine-tuning along with inference for 7B models task 26 hours. Computing calibration parameters takes around 30 minutes to 2 hours for each method.

A.2 SUPER-NATURALINSTRUCTIONS

We evaluate models on a subset of 279 tasks from the SUPER-NATURALINSTRUCTIONS benchmark (Wang et al., 2022), obtained from `https://github.com/allenai/natural-instructions`. We use up to 1000 evaluation examples for each task. Altogether, our evaluation set consists of 264,176 examples.

SUPER-NATURALINSTRUCTIONS is a benchmark containing instances from many individual datasets, the license of each is detailed in `https://github.com/allenai/natural-instructions` next to the task's files.

A.3 LoRA Hyperparameters

We use the same LoRA hyperparamets used by Dettmers et al. (2023) for fine-tuning on SUPER-NATURALINSTRUCTIONS, except we use bf16 training instead of 8bit, a warmup rate of 0.0, and 5 epochs. Specifically, we use a learning rate of 0.002, LoRA $r = 64$ and LoRA $\alpha = 16$.

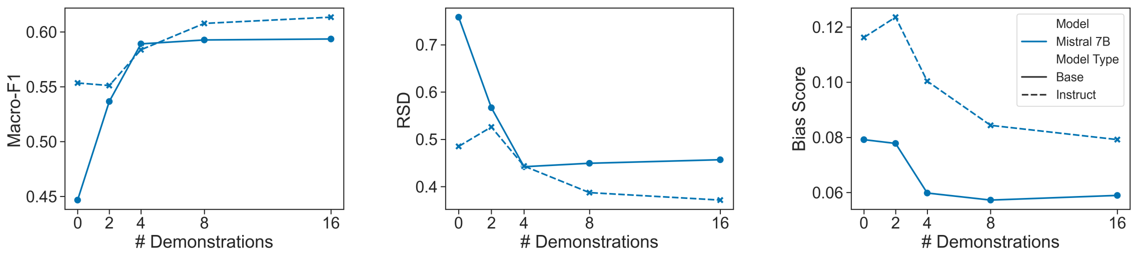
B Additional Results

B.1 Label Bias in LLMs

For results on Mistral and Falcon models before the application of any mitigation approaches, see Fig. 5 and Fig. 6 respectively.

B.2 Mitigation Approaches

For full results on Mistral and Falcon models including all mitigation methods, see Fig. 7 and Fig. 8 respectively.

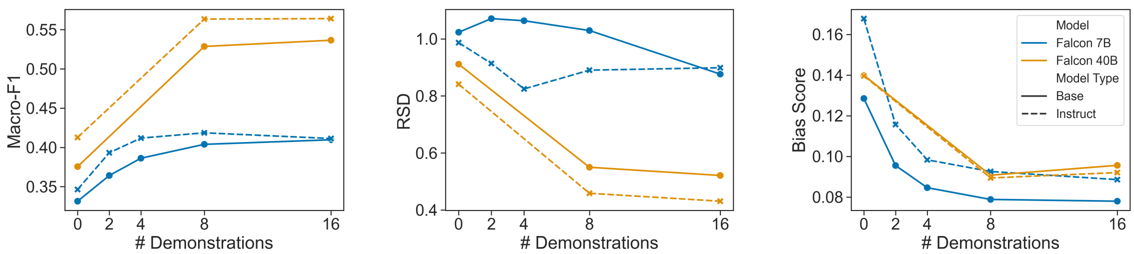


(a) Performance (Macro-F1)

(b) Label bias (*RSD*)

(c) Label bias (*BiasScore*)

Figure 5: Performance and label bias metrics for Mistral 7B pretrained and instruction-tuned models.

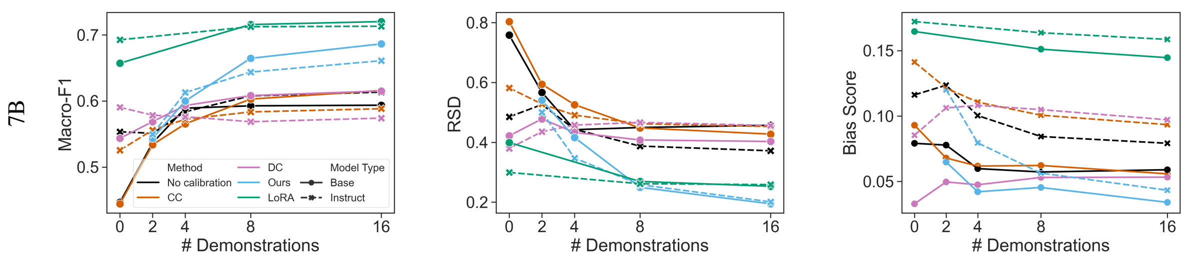


(a) Performance (Macro-F1)

(b) Label bias (*RSD*)

(c) Label bias (*BiasScore*)

Figure 6: Performance and label bias metrics for Falcon pretrained and instruction-tuned models (7B/40B).

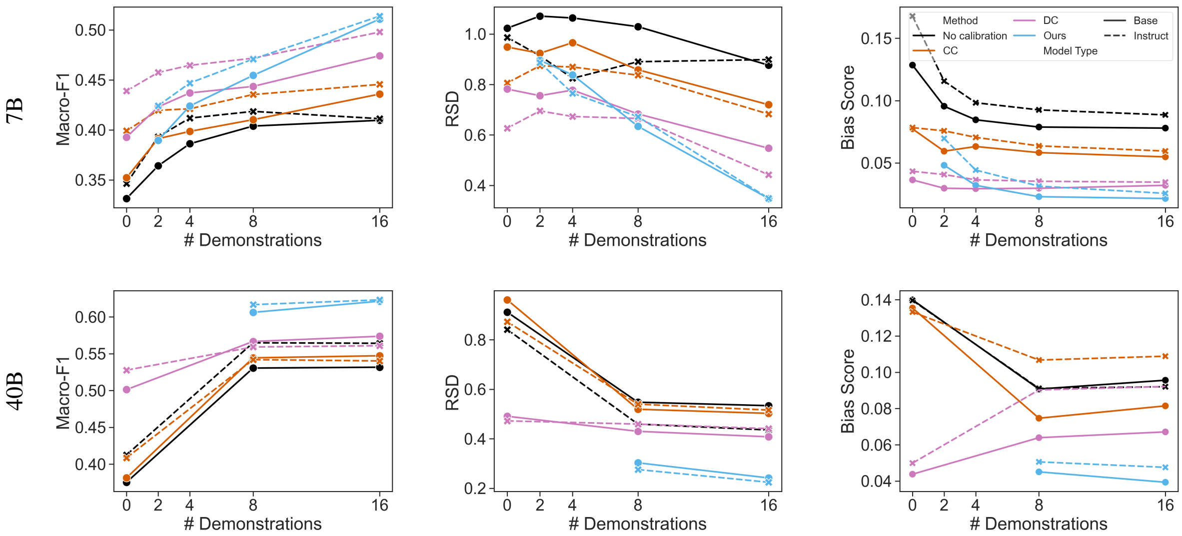


(a) *Macro-F1*

(b) *RSD*

(c) *Bias Score*

Figure 7: The effect of label bias mitigation methods on performance and bias for Mistral models.



(a) *Macro-F1* (b) *RSD* (c) *Bias Score*

Figure 8: The effect of label bias mitigation methods on performance and bias for Falcon models.