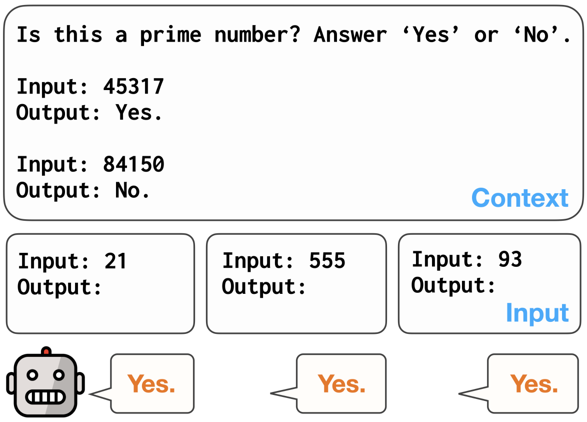
Beyond Performance: Quantifying and Mitigating Label Bias in LLMs

Yuval Reif **Roy Schwartz**
School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel
{yuval.reif,roy.schwartz1}@mail.huji.ac.il

Abstract

Large language models (LLMs) have shown remarkable adaptability to diverse tasks, by leveraging 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. Our results emphasize that label bias in the predictions of LLMs remains a barrier to their reliability.[1]



```
Is this a prime number? Answer 'Yes' or 'No'.

Input: 45317
Output: Yes.

Input: 84150
Output: No.
```

```
Input: 21
Output:
```

```
Input: 555
Output:
```

```
Input: 93
Output:
```

Figure 1: LLMs exhibit *label bias*—a tendency to output a given label regardless of the context and input (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.

1 Introduction

Large language models (LLMs) have demonstrated impressive 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).

Despite these efforts, label bias evaluation relies on *performance* metrics such as accuracy, rather than metrics designed to directly quantify 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 directly measure the label bias in model predictions, which we derive from previous work on fairness and label bias esti-

[1] We release our code at https://github.com/schwartz-lab-NLP/label-bias.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6784–6798
June 16-21, 2024 ©2024 Association for Computational Linguistics

mation, we evaluate ten LLMs on 279 diverse classification and multiple-choice tasks from SUPER-NATURALINSTRUCTIONS (Wang et al., 2022). We examine both performance and bias along axes such as scale and number of in-context 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 more accurately estimates a model's label bias, using only its predictions on the in-context demonstrations. Compared to existing LLM bias 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 when evaluating their performance. Moreover, adjusting models to their tasks through more accurate and effective estimation of biases holds promise 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, providing 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), briefly outline methods to mitigate it (§2.2), and finally review approaches to evaluate label bias as well as define the metrics we use in this work (§2.3).

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 $[(x^1, y^1), \ldots, (x^k, y^k)]$, 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 Bias Mitigation

The predominant approach to alleviate label bias is to calibrate the model's output probabilities post-hoc, for a specific context prompt C. Such methods typically first estimate the model's label bias using its output probabilities on a set of inputs, which can be content-free (e.g., "N/A" or random words from the task's domain; Zhao et al. 2021; Fei et al. 2023) or ordinary task inputs (Han et al., 2023). Next, calibration parameters are chosen based on this estimate, and used to adjust the original output probabilities during inference to generate the (hopefully unbiased) output.

2.3 Evaluation Measures

Most LLM label bias analysis relies on indirect assessments. For instance, some work inspected 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 which 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, e.g., assigning the label "yes" with an *average* output probability of 0.55, while "no" with 0.45; and through the model's predictions for different labels, e.g., achieving a recall of 0.50 for instances labeled "yes", compared to 0.40 on "no" (Mehrabi et al., 2021). Below we describe methods to measure each of these notions of bias.

Probabilistic approach 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 models at larger scales. Recently, Fei et al. (2023) proposed to measure a model's label bias by considering two sets of inputs: a set of synthetic, content-free task inputs $\hat{X}_{cf}$, and inputs consisting of random vocabulary words $\hat{X}_{rand}$. For each input, they compute the output probabilities on every label $y \in Y$, and finally compute the model's *mean* predicted probabilities across both sets, $\hat{p}_{cf}$ and $\hat{p}_{rand}$:

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

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

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

Importantly, since Fei et al. (2023) also use the model's predictions on the content-free inputs $\hat{X}_{cf}$ to calibrate it, 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 synthetic inputs, we use in-distribution examples held-out from the test set, $\hat{X}_{i.d.} = ((x_1, y_1), \ldots, (x_m, y_m))$. This setup allows to account for label imbalance in the data used for bias estimation $\hat{X}_{i.d.}$, 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 $\ell \in Y$, $\hat{X}^{\ell}_{i.d.} = \{(x, y) \in \hat{X}_{i.d.} \mid y = \ell\}$, by computing:

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

and then set $\hat{p}_{i.d.}$ 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 considered as the "ideal" and unbiased mean output distribution (Zhao et al., 2021). Finally, we define the model's **bias score** as the total variation distance between these two distributions:

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

Outcomes-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 class-wise accuracy $(acc_1, \ldots, acc_{|Y|})$, 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 on more difficult instances. We therefore suggest reporting both measures.

[2] In case examples for an infrequent label $\ell \in Y$ are not found in $\hat{X}_{i.d.}$, we exclude it from the computation of $\hat{p}_{i.d.}$.

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

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 "yes/no" or "A/B/C". We sample 1,000 evaluation examples for all tasks with larger data sizes, and additionally sample 32 held-out examples for computing the bias score metric (§2.3), and 64 more examples to use 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. For details on the selected tasks, see App. B.

3.2 Models and Evaluation Setup

We experiment with models of different sizes from three LLM 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 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 to compute the output probabilities (Holtzman et al., 2021). For additional implementation details, see App. A.

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). As roughly 20% of *Flan* consists of examples from SUPER-NATURALINSTRUCTIONS, our evaluation of Llama-2 instruction-tuned models is likely affected by data contamination (Magar and Schwartz, 2022). Still, our results show both 7B and 13B chat 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 exact instances 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

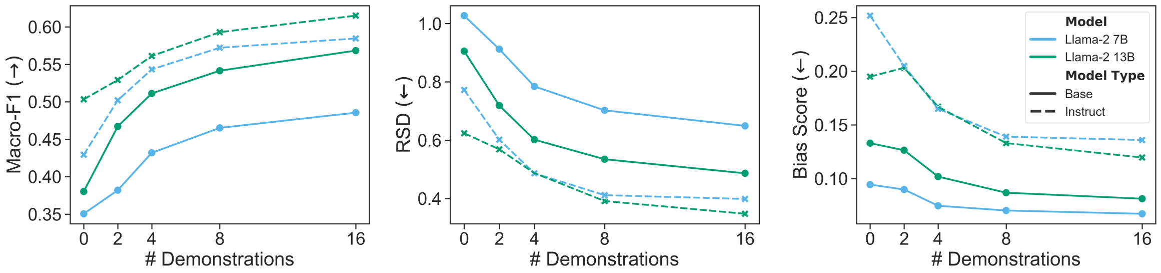
We evaluate the effects of three label bias mitigation methods: two calibration methods designed to correct a model's label bias by adjusting its output scores; and few-shot LoRA fine-tuning (Hu et al., 2022), which adapts the model to the task and its label distribution. We describe the methods below.

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 during inference to obtain calibrated, debiased probabilities $q = \text{softmax}(Wp)$. To determine the calibration parameters W, they first estimate the bias by computing the model's average predicted probabilities $\hat{p}$ on a small set of "placeholder" content-free input strings, such as "N/A", which replace the ordinary task input that follows C.[4] Finally, they set $W = \text{diag}(\hat{p})^{-1}$, which ensures that the output class probabilities for the average content-free input are uniform, aiming to reduce bias on unseen examples.

Domain-context calibration (DC) Following CC, Fei et al. (2023) proposed to estimate and mitigate the label bias arising from the textual distribution of the task's domain, by using task-specific content-free inputs to compute $\hat{p}$. They construct such 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 20 times, and set $\hat{p}$ to be the average output probabilities over all examples. Given a test example with original output probabilities p, they then use the calibrated probabilities $q = \text{softmax}(p/\hat{p})$.

Few-shot fine-tuning Finally, we experiment with few-shot, parameter-efficient fine-tuning 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

[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* is substantially worse after instruction tuning and does not improve when scaling models up in most evaluated settings.

computational constraints, we only run LoRA on Llama-2 7B and Mistral 7B, only consider values of $k \in (0, 8, 16)$, and average across two sets of demonstrations. See App. A for more 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 (see App. C.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 metrics—*RSD* (Fig. 2b) and *BiasScore* (Fig. 2c). We observe that label bias is substantial across most evaluated settings: When prompted with two or no demonstrations, all models obtain high *RSD* values of 0.6 or more, with base models obtaining even higher values around 0.9. This implies a widespread disparity in model performance across classes in many of the evaluated tasks, and indicates that for most tasks, models primarily succeed on instances of certain classes, while consistently failing on others. Increasing the number of demonstrations to 8 helps reduce the bias, but *RSD* remains substantial at around 0.4, and adding further demonstrations results in little to no improvement.

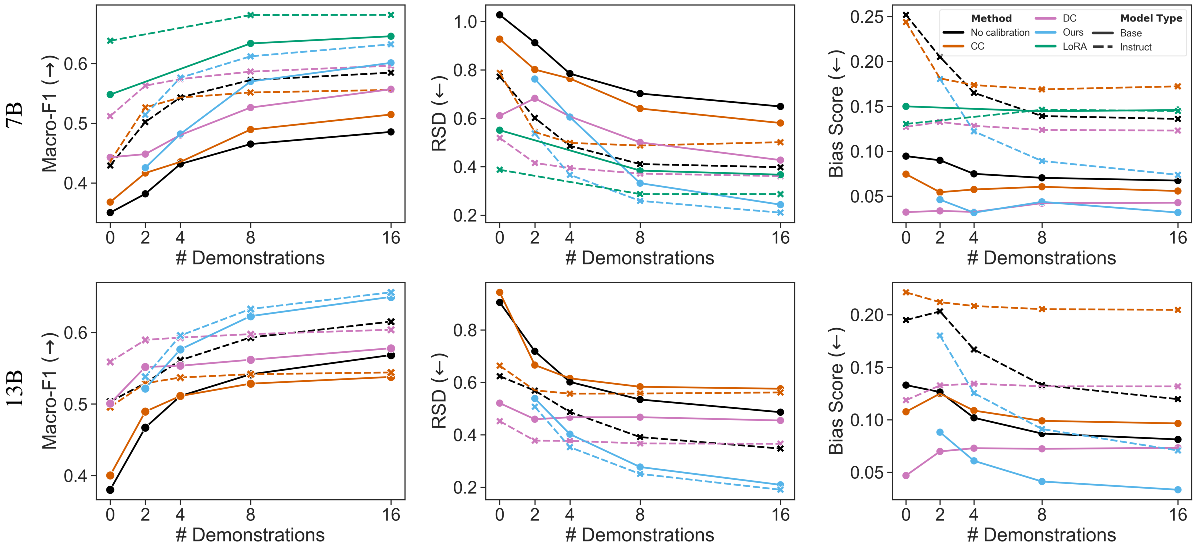
Similarly, we find *BiasScore* improves considerably when using sufficient demonstrations, with models obtaining values as high as 0.25 when using no demonstrations, to around 0.05 for the best model and setting. High *BiasScore* values indicate the model is uncalibrated, and tends to make overly confident predictions on certain labels regardless of the input. Although *BiasScore* can be relatively small for some models—indicating their average output distribution is close to uniform—when observed together with high *RSD*, it implies that the model subtly but persistently assigns more probability mass to the preferred labels, resulting in substantial bias.

4.2 Differences between the Bias Measures

We further observe 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 base counterparts. Similarly, while *RSD* improves with scaling, the *BiasScore* of smaller models is lower.

We note that higher performance together with lower *RSD* means that the model's performance has improved across most classes. In contrast, higher *BiasScore* indicates that its average predicted probabilities grew farther than uniform. Taken together, this implies that the scaled-up and instruction-tuned models are making more confident predictions on some classes, but not on others. This could mean more confident correct predictions on the preferred classes, or more confidently wrong predictions on others (or both). Altogether, this suggests 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 measurement of different aspects of label bias. We hence use both in further experiments to provide a more comprehensive understanding of label bias in model predictions.



(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 bias for $k \in \{8, 16\}$.

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, and observe similar trends for other models (App. C.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, especially when using few or no in-context demonstrations, where baseline performance is relatively low. However, when calibrating instruction-tuned models prompted with a 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, here we observe that for the two calibration methods, changes in both *RSD* and *BiasScore* are correlated with changes in performance. We find that CC substantially worsens label bias in instruction-tuned models, and can also increase bias for base models. Conversely, while DC alleviates bias in many of the evaluated settings, it is largely unsuccessful in mitigating it when prompting instruction-tuned models with 8 or more demonstrations. LoRA proves effective for improving *RSD* in all settings, but *RSD* values still remain relatively high. In contrast, *BiasScore* noticeably increases after LoRA 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 considerably more computationally expensive than calibration.

5 Mitigating Label Bias by Calibrating on Demonstrations

Motivated by the challenges 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 such difficulties is that the inputs used for calibration in CC and DC are very distinct from the more curated, high-quality inputs models observe during instruction-tuning (Touvron et al., 2023).[5]

Seeking to use more naturally-occurring inputs, and to avoid any reliance on additional held-out examples, we propose to calibrate models using the in-context examples readily available in few-shot prompts. We therefore need to obtain the model's output probabilities on these inputs to estimate its bias. However, as these examples appear alongside their labels in the context provided to the model, it could simply copy the correct answer from the prompt, leading to unreliable bias estimates. We introduce a simple method to alleviate this concern.

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 it 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), resulting in $k-1$ demonstrations.[6] We thus obtain k output probabilities:

$$p^i(y) = p(y \mid x^i, C_{-i})$$

To reliably estimate $\hat{p}$, we further need to account for the demonstrations' labels y^i: for imbalanced choices of demonstrations (e.g., class imbalance), 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* (§2.3): We first average p^i's associated with the same label ℓ, $\mathcal{D}_\ell = \{p^i \mid y^i = \ell\}$, and then set $\hat{p}$ as the mean of these intra-label averages:

$$\hat{p}(y) = \frac{1}{|Y|} \sum_{\ell \in Y} \left(\frac{1}{|\mathcal{D}_\ell|} \sum_{y^i \in \mathcal{D}_\ell} p^i(y) \right)$$

Finally, we use $\hat{p}$ 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 of §4.3. We report our results for Llama-2 models in Fig. 3 (**cyan** lines), finding similar trends in other models (App. C.2). Comparing our method to other calibration approaches, we find LOOC surpasses CC and DC by a wide margin in both performance and bias metrics for prompts with $k = 8, 16$ demonstrations. Importantly, using LOOC to calibrate instruction-tuned models in this setting dramatically improves upon the uncalibrated model, whereas other calibration methods fail to achieve meaningful gains (§4.3). Further, LOOC nearly closes the gap with LoRA-level performance while improving upon it in both bias metrics, yet uses substantially less compute.

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 while our method can substantially reduce label bias compared to other approaches, the remaining *RSD* is still considerable, indicating that model performance is still biased on some tasks.

6 Analysis

We study the effect of different factors on the extent of label bias in model predictions: the semantic meaning of the task labels (§6.1), the level of label imbalance in the demonstrations (§6.2), and the choice of demonstrations (§6.3).

6.1 Semantically Equivalent Labels

The output space for classification tasks often consists of labels with strong semantic meaning (e.g., "Positive" vs. "Negative"). Recent work has indicated that, when faced with such labels, models are affected by semantic priors from their pretraining or instruction-tuning (Wei et al., 2023; Min et al., 2022b) that could affect label bias (Fei et al., 2023).

We examine whether models exhibit lower label bias when the task's labels are semantically equivalent and interchangeable. 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 models choose a logical continuation for an input text between two options, usually labeled A and B. This results in 18 tasks with semantically equivalent labels.

We compare label bias on this subset of tasks and the entire evaluation suite for Llama-2 models in Fig. 4, with results for other models largely fol-

[5] 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.

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

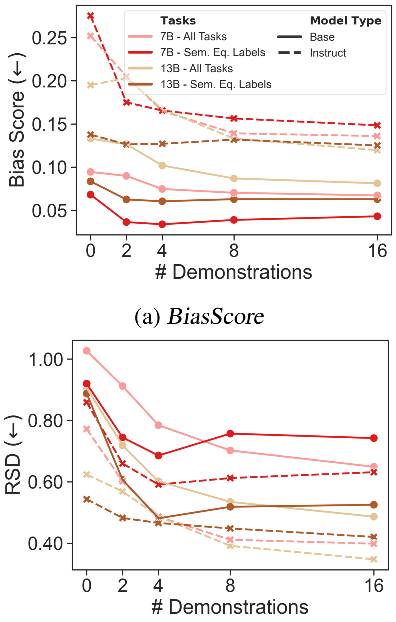
lowing similar trends (App. C.3). We find that, in most cases, models demonstrate lower label bias on tasks with semantically equivalent labels. This is especially evident in settings with few or no demonstrations, where models are typically strongly biased (§4.1). Still, *RSD* levels for such tasks remain relatively high across all evaluated settings. Further, we observe that instruction-tuned models prompted with 8 or more demonstrations are often *more* biased on this subset of tasks. In summary, although using semantically equivalent labels may potentially mitigate bias in scenarios with limited demonstrations, LLMs still exhibit substantial label bias when faced with such labels.

6.2 Imbalanced In-context Demonstrations

Label imbalance in the in-context demonstration set was previously shown to amplify label bias (Zhao et al., 2021) as well as decrease model performance (Min et al., 2022a), but such results were derived on a restricted set of tasks. We use our evaluation suite to investigate the observed label bias and performance of models when varying the level of imbalance in the demonstrations. To establish a consistent definition of label imbalance across different tasks, we use the subset of binary classification tasks ($N = 197$) with $k = 8$ demonstrations. Given a task with labels $L = \{\ell_A, \ell_B\}$ and a context C, we define $p_\uparrow$ as the proportion of the most frequent label in the demonstrations of C, such that $p_\uparrow$ attains values in $\{0.5, 0.625, 0.75, 0.875, 1.0\}$. Specifically, $p_\uparrow = 0.5$ means the labels are perfectly balanced, and $p_\uparrow = 1.0$ means the demonstrations only include examples for one of the labels.

For every task, we prompt Llama-2 (7B/13B) and Mistral (7B) models using 10 different sets of demonstrations, with 2 sets for each value of $p_\uparrow$: one where ℓ_A is the most frequent label in C, and another where ℓ_B is the most frequent, as well as two different balanced sets ($p_\uparrow = 0.5$).[7] We group measurements taken across different tasks and demonstration sets by their level of label imbalance $p_\uparrow$, and inspect the average results per level.

We report our results in Fig. 5. Examining the two bias metrics, *RSD* (Fig. 5a) and *BiasScore* (Fig. 5b), we observe that both pretrained and instruction-tuned models are resistant to label imbalance: Increased imbalance does not result in notable gains in bias, unless the imbalance is very extreme—specifically, when the demonstrations include only a single or no demonstrations for one of the labels ($p_\uparrow > 0.75$). Interestingly, model performance follows the same trends (Fig. 5c). Overall, our results indicate that for most tasks, the impact of label imbalance in the demonstrations set is minimal, except for cases of severe imbalance.



(a) *BiasScore*

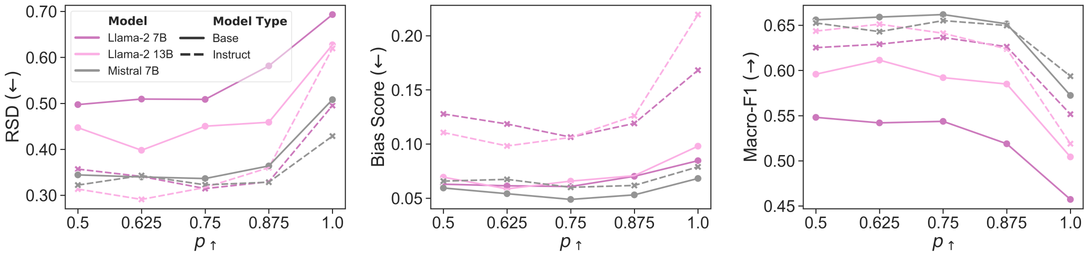
(b) *RSD*

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.

6.3 Choice of Demonstrations

The performance of LLMs in in-context learning was shown to be sensitive to the exact choice of demonstrations used to prompt the model (Liu et al., 2022; Chang and Jia, 2023). We examine whether such choices also impact the extent of *label bias* in model predictions. We assess the performance and bias of Llama-2 (7B/13B) and Mistral (7B) models across 5 different sets of $k = 8$ demonstrations for each task in our evaluation suite. In addition to reporting the mean and standard deviation of each metric, we use several oracle methods to aggregate and choose a specific demonstration set per task when computing the overall cross-task performance and bias metrics. Specifically, we select the demonstration sets that attain the following, per task: *best performance*; *worst performance*;

[7] To build each set, we randomly select and permute 8 demonstrations from a pool of 16 held-out examples, while controlling for the selected number of examples per label.



(a) *RSD* (b) *BiasScore* (c) *Macro-F1*

Figure 5: Label bias and performance metrics for Llama-2 (7B/13B) and Mistral (7B) models, when aggregated by the level of imbalance in the demonstrations set used for prompting the model, measured by the proportions of its most frequent label ($p_\uparrow$). For most tasks, label imbalance has only minor impact on both bias and performance, unless the imbalance is extreme. Instruction-tuned models are less sensitive to imbalance.

Choice of Demonstrations	**F1** (↑)	***RSD*** (↓)	***BiasScore*** (↓)
Mean (SD)	0.47 (± 0.088)	0.69 (± 0.316)	0.077 (± 0.039)
Best Performance	0.565	0.384	0.068
Median Performance	0.478	0.656	0.079
Worst Performance	0.355	1.083	0.071
Least Bias – *by RSD*	0.553	0.35	0.066
Least Bias – *by BiasScore*	0.457	0.755	0.024
Most Bias – *by RSD*	0.358	1.102	0.069
Most Bias – *by BiasScore*	0.436	0.781	0.119

Table 1: Results of Llama-2 7B base model when prompted with 5 different sets of demonstration on our evaluation suite. We employ oracles to aggregate and compute cross-task results by choosing a specific set of demonstrations for each task. Label bias is highly sensitive to the choice of in-context examples.

median performance; *least bias*; and *most bias*.

We report our results for Llama-2 7B base in Tab. 1, with other models showing similar trends (App. C.4). We find that label bias, similarly to model performance, is highly sensitive to the choice of demonstrations, as indicated by the high variance across sets. Interestingly, the set of demonstrations that attains the worst performance also leads to strong bias, and vice-versa. In fact, we find that performance and bias are anti-correlated, with strong Pearson correlation for *RSD* ($r = -0.74$) and moderate for *BiasScore* ($r = -0.30$), indicating that when LLMs underperform in classification, it is often due to prompts that exacerbate bias. We leave further research into demonstrations that lead to biased and unbiased predictions to future work.

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 exhibit positional bias when presented with several texts for evaluation and ranking. Pezeshkpour and Hruschka (2023) and Zheng et al. (2024) exposed a similar bias in multi-choice QA. Si et al. (2023) studied inductive biases in in-context learning. Complimentary to these works, we study label bias and seek to improve its evaluation and mitigation.

Calibrating Label Bias in LLMs Recent work introduced calibration methods to mitigate label bias in LLMs (Zhao et al., 2021; Fei et al., 2023). Han et al. (2023) proposed to fit a Gaussian mixture to the model's output probabilities and use it for calibration, but their approach requires hundreds of labeled examples. Concurrently to our work, Jiang et al. (2023b) proposed to generate inputs for calibration by conditioning models on the context prompt, and Zhou et al. (2023) calibrate models using model output probabilities on the entire test set. While the motivation for both methods is similar to ours, our approach does not require access to the test set, or any compute to obtain inputs for calibration. Importantly, unlike previous work on bias calibration, our main focus is the evaluation of label bias in LLMs.

8 Conclusion

The label bias of LLMs severely hinders their reliability. We considered different approaches for quantifying this bias. Through extensive experiments with ten LLMs across 279 classification tasks, we found that substantial amounts of label bias exist in LLMs. Moreover, we showed this bias persists as LLMs increase in scale, are instruction-tuned, are provided in-context examples, 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 better estimate and mitigate LLM biases.

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). Therefore, the extent to which our findings apply to such models is unclear.

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.

Acknowledgements

This work was supported in part by the Israel Science Foundation (grant no. 2045/21).

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*.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144, Toronto, Canada. Association for Computational Linguistics.

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)*.

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.

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.

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

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

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

Our implementation and pretrained model checkpoints use the Huggingface Transformers library (Wolf et al., 2020). Our code for model evaluation on SUPER-NATURALINSTRUCTIONS is based on the code from Wang et al. (2023b).

Inference When running inference, we load all models using bf16, except for Falcon-40B, which we load using 8-bit inference. We evaluate models using a maximum sequence length of 1024. When incorporating in-context demonstrations into the prompt, the demonstrations are added one by one until the maximal sequence length is reached, while ensuring enough space remains for the input of the evaluated example. Any remaining demonstrations exceeding this length are excluded from the prompt. Consequently, when evaluating tasks with k demonstrations, the contexts for tasks with very long inputs may contain fewer than k demonstrations. In our experiment detailed in §6.2, which investigates the impact of label imbalance in the demonstrations set on label bias, we use a sequence length of 2048 and only analyze results for tasks where the prompt contains precisely k demonstrations, excluding other instances from our reported findings.

Compute 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 takes 26 hours. Computing calibration parameters, including running inference on inputs required for calibration, takes around 30 minutes to 2 hours for each model, depending on the method used.

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

B Evaluation Suite

We evaluate models on a subset of 279 tasks from the SUPER-NATURALINSTRUCTIONS benchmark (Wang et al., 2022). We use up to 1000 evaluation examples for each task. Altogether, our evaluation set consists of 264,176 examples.

We detail the categories of the selected tasks along with the number of tasks corresponding to each category in Tab. 2. We also report the distribution of the number of labels across tasks in Tab. 3, as well as the 20 most frequent labels in Tab. 4.

C Supplementary Results

C.1 Performance and Label Bias

We provide additional results for the performance and label bias of models (§4.1) for Mistral (Fig. 8) and Falcon (Fig. 9) models.

C.2 Bias Mitigation Methods

We present additional results for the impact of bias mitigation methods (§4.3) for Mistral (Fig. 10) and Falcon (Fig. 11) models.

C.3 Semantically Equivalent Labels

We present additional results for the analysis on label bias for tasks with semantically equivalent labels (§6.1) for Mistral (Fig. 6) and Falcon (Fig. 7) models.

C.4 Choice of Demonstrations

We present additional results for the analysis on the sensitivity of label bias to the choice of in-context examples (§6.3). We report separate results for each model: Llama-2 7B chat (Tab. 5), Llama-2 13B base (Tab. 6), Llama-2 13B chat (Tab. 7), Mistral 7B base (Tab. 8), and Mistral 7B instruct (Tab. 9).

Task Category	**# of Tasks**	**# of Instances**
Sentiment Analysis	39	37748
Text Categorization	30	27652
Toxic Language Detection	25	24114
Commonsense Classification	23	22239
Textual Entailment	15	14613
Question Answering	13	12380
Answerability Classification	12	11286
Text Matching	11	10807
Question Understanding	8	7730
Text Completion	7	7000
Speaker Identification	6	4739
Ethics Classification	6	5501
Text Quality Evaluation	6	6000
Dialogue Act Recognition	6	5401
Stereotype Detection	6	5627
Cause Effect Classification	5	4200
Word Relation Classification	5	4680
Gender Classification	5	5000
Negotiation Strategy Detection	5	4150
Coherence Classification	5	5000
Answer Verification	4	4000
Information Extraction	4	4000
Dialogue State Tracking	3	2855
Coreference Resolution	3	3000
Linguistic Probing	2	1808
Pos Tagging	2	2000
Irony Detection	2	1933
Word Semantics	2	1210
Text to Code	2	2000
Intent Identification	2	2000
Section Classification	2	2000
Tasks With Unique Categories	13	11503
Total	279	264,176

Table 2: Categories of tasks included in our evaluation suite, based on SUPER-NATURALINSTRUCTIONS, along with the number of tasks per category and the total number of instances used for evaluating models.

Answer Choices	**Number of Tasks**
2	39
3	30
4	25
5	23
6-9	14
10+	8

Table 3: Distribution of the number of labels across tasks in our evaluation suite.

Label	**Freq.**	**Label**	**Freq.**
no	76	3	14
yes	75	negative	13
1	31	4	10
true	20	c	9
false	20	5	9
b	19	neutral	8
a	19	d	7
2	18	anti-stereotype	5
0	17	stereotype	5
positive	14	pos	5

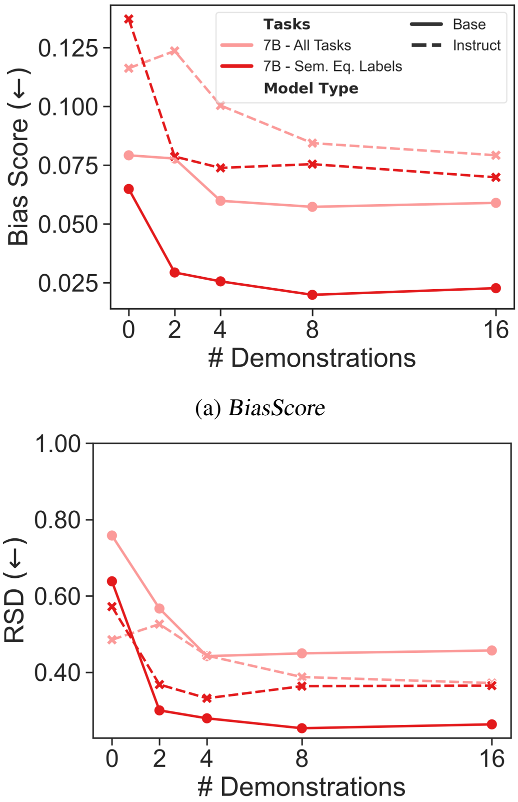
Table 4: The 20 most frequent labels in our evaluation suite and the number of tasks they appear in.

Choice of Demonstrations	**F1** (↑)	***RSD*** (↓)	***BiasScore*** (↓)
Mean (SD)	0.571 (± 0.059)	0.417 (± 0.178)	0.141 (± 0.061)
Best Performance	0.636	0.267	0.095
Median Performance	0.577	0.401	0.137
Worst Performance	0.494	0.613	0.197
Least Bias – *by RSD*	0.619	0.216	0.084
Least Bias – *by BiasScore*	0.614	0.25	0.07
Most Bias – *by RSD*	0.503	0.645	0.205
Most Bias – *by BiasScore*	0.511	0.613	0.22

Table 5: Results of Llama-2 7B chat model when prompted with 5 different sets of demonstration on our evaluation suite. We employ oracles to aggregate and compute cross-task results when choosing a specific set of demonstrations for each task.

Choice of Demonstrations	**F1** (↑)	***RSD*** (↓)	***BiasScore*** (↓)
Mean (SD)	0.54 (± 0.069)	0.546 (± 0.205)	0.088 (± 0.031)
Best Performance	0.618	0.352	0.065
Median Performance	0.544	0.522	0.084
Worst Performance	0.452	0.782	0.114
Least Bias – *by RSD*	0.605	0.314	0.062
Least Bias – *by BiasScore*	0.592	0.369	0.052
Most Bias – *by RSD*	0.457	0.806	0.118
Most Bias – *by BiasScore*	0.475	0.747	0.128

Table 6: Results of Llama-2 13B base model.



(a) *BiasScore*

(b) *RSD*

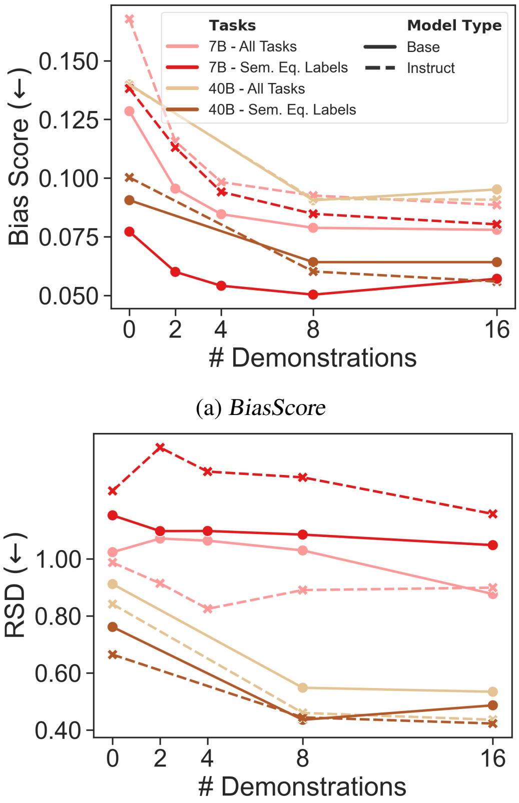
Figure 6: Label bias metrics for Mistral 7B models when evaluated on all tasks in our evaluation suite (*All*) vs. a subset of tasks with semantically equivalent labels (*Sem. Eq. Labels*).

Choice of Demonstrations	F1 (↑)	*RSD* (↓)	*BiasScore* (↓)
Mean (SD)	0.592 (± 0.058)	0.397 (± 0.173)	0.134 (± 0.058)
Best Performance	0.656	0.241	0.092
Median Performance	0.597	0.383	0.13
Worst Performance	0.517	0.584	0.185
Least Bias – *by RSD*	0.643	0.201	0.085
Least Bias – *by BiasScore*	0.632	0.251	0.067
Most Bias – *by RSD*	0.523	0.622	0.194
Most Bias – *by BiasScore*	0.534	0.58	0.21

Table 7: Results of Llama-2 13B chat model.

Choice of Demonstrations	F1 (↑)	*RSD* (↓)	*BiasScore* (↓)
Mean (SD)	0.601 (± 0.092)	0.432 (± 0.239)	0.064 (± 0.034)
Best Performance	0.692	0.225	0.057
Median Performance	0.616	0.387	0.064
Worst Performance	0.47	0.747	0.064
Least Bias – *by RSD*	0.68	0.196	0.055
Least Bias – *by BiasScore*	0.579	0.484	0.021
Most Bias – *by RSD*	0.477	0.77	0.067
Most Bias – *by BiasScore*	0.563	0.537	0.102

Table 8: Results of Mistral 7B base model.



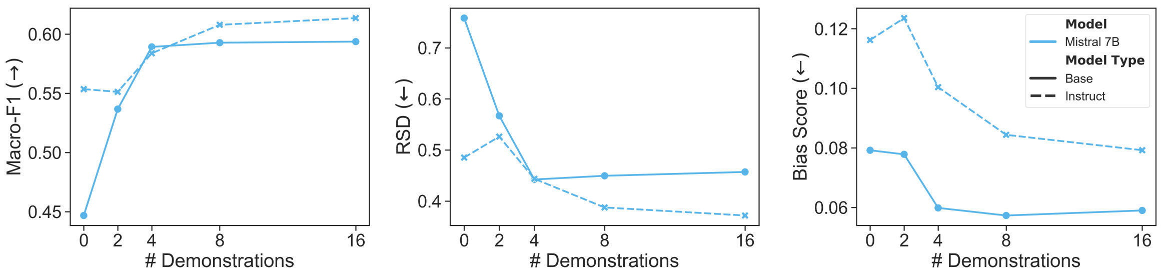
(a) *BiasScore*

(b) *RSD*

Figure 7: Label bias metrics for Falcon (7B/40B) models when evaluated on all tasks in our evaluation suite (*All*) vs. a subset of tasks with semantically equivalent labels (*Sem. Eq. Labels*).

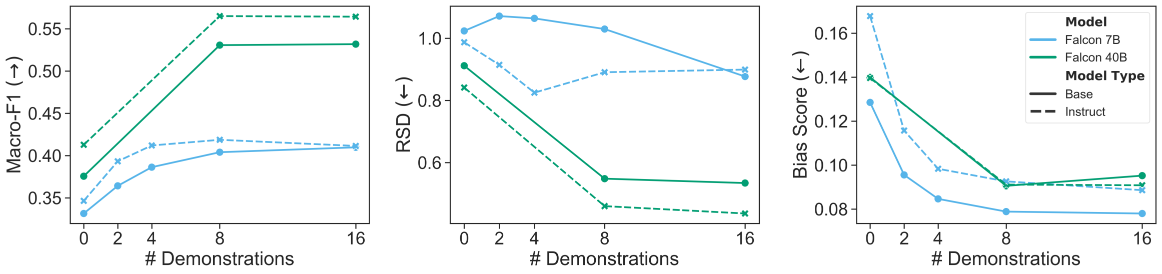
Choice of Demonstrations	F1 (↑)	*RSD* (↓)	*BiasScore* (↓)
Mean (SD)	0.607 (± 0.059)	0.389 (± 0.167)	0.085 (± 0.035)
Best Performance	0.672	0.232	0.06
Median Performance	0.613	0.368	0.08
Worst Performance	0.53	0.585	0.115
Least Bias – *by RSD*	0.663	0.202	0.059
Least Bias – *by BiasScore*	0.653	0.245	0.044
Most Bias – *by RSD*	0.535	0.604	0.119
Most Bias – *by BiasScore*	0.553	0.545	0.131

Table 9: Results of Mistral 7B instruct model.



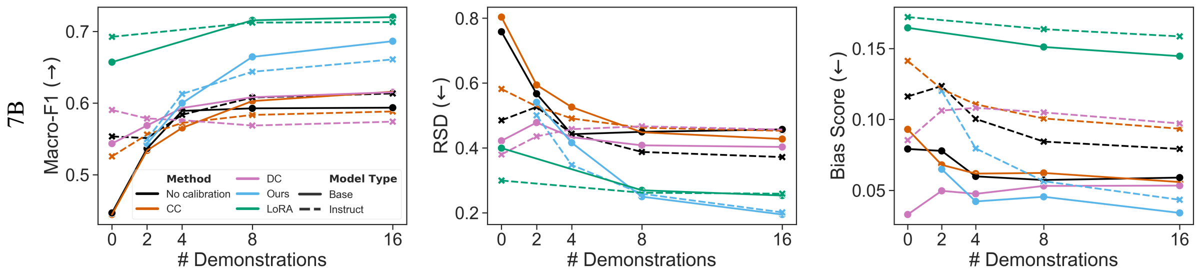
(a) Performance (Macro-F1) (b) Label bias (*RSD*) (c) Label bias (*BiasScore*)

Figure 8: Performance and label bias metrics for Mistral 7B pretrained and instruction-tuned models.



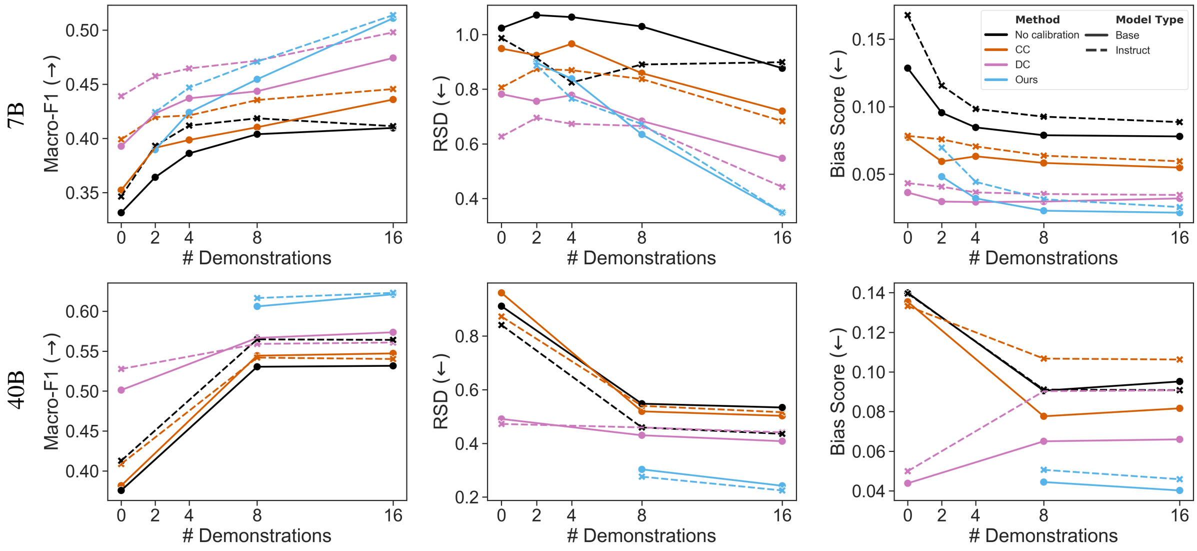
(a) Performance (Macro-F1) (b) Label bias (*RSD*) (c) Label bias (*BiasScore*)

Figure 9: Performance and label bias metrics for Falcon pretrained and instruction-tuned models (7B/40B).



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

Figure 10: The effect of label bias mitigation methods on performance and bias for Mistral models.



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

Figure 11: The effect of label bias mitigation methods on performance and bias for Falcon models.