

Mitigating Label Length Bias in Large Language Models

Mario Sanz-Guerrero¹ and Katharina von der Wense^{1,2}

¹Johannes Gutenberg University Mainz, Germany

²University of Colorado Boulder, USA

{msanz, k.vonderwense}@uni-mainz.de

Abstract

Large language models (LLMs) are powerful zero- and few-shot learners. However, when predicting over a set of candidate options, LLMs suffer from label biases, and existing calibration methods overlook biases arising from multi-token class labels. We tackle an issue we call *label length bias*, where labels of different lengths are treated inconsistently, even after standard length normalization. To mitigate it, we propose *normalized contextual calibration* (NCC), an effective method that normalizes and calibrates predictions at the full-label level. NCC achieves statistically significant improvements over prior approaches across multiple datasets and models, with gains of up to 10% F1. Moreover, NCC extends bias mitigation to broader tasks such as multiple-choice question answering. Our analysis shows that, when combined with in-context learning, NCC is less sensitive to few-shot example selection, requires fewer examples for competitive performance, and produces more reliable confidence estimates. These findings highlight the importance of mitigating full-label biases to improve the performance and robustness of LLM-based methods, particularly in real-world applications where class labels naturally consist of multiple tokens.

1 Introduction

Large language models (LLMs) have demonstrated strong zero- and few-shot capabilities, enabling them to perform new tasks with little or no task-specific supervision. One of the most common ways to leverage these capabilities is through in-context learning (ICL), where models are prompted with a small set of labeled examples to guide their predictions (Brown et al., 2020).

A key challenge when using LLMs to predict over a set of candidate options (e.g., for text classification tasks) is label bias, where certain labels are inherently favored due to their lexical properties

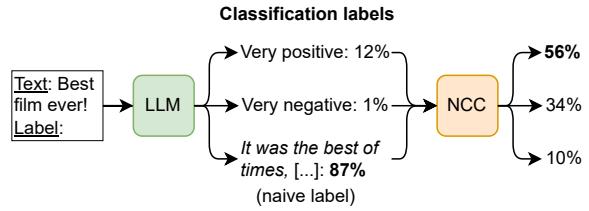


Figure 1: NCC enables the use of calibration in multi-token tasks, mitigating label biases of LLMs and improving their performance. The figure shows real numbers obtained with the Llama 3.1 (8B) model.

or prior likelihoods: consider a sentiment analysis task where a model must decide between the two classes “*good*” (the positive class) and “*atrocious*” (the negative class) – the model would be biased toward generating “*good*”, as it has seen this word more frequently in its training data. Zhao et al. (2021) highlights the need for calibration to ensure predictions reflect input alignment rather than model-internal biases. However, existing calibration techniques are insufficient for multi-token class labels, as they typically consider only single-token labels. This approach ignores the joint probability of full-label sequences, which are frequently found in practical scenarios.

In this work, we spotlight a previously overlooked type of bias in LLMs: *label length bias*, where models are biased toward certain classes based on label lengths rather than on semantic alignment with the input. To mitigate this bias, we propose *normalized contextual calibration* (NCC), a novel calibration method that accounts for full-label probabilities. NCC first normalizes each label’s multi-token probability by its length. It then calibrates the model’s predictions by dividing each normalized probability by the corresponding prior probability – computed from content-free inputs – to explicitly adjust for label biases.

We evaluate NCC across multiple datasets and models. Our results show that **NCC consistently**

outperforms previous approaches, achieving statistically significant improvements of up to 10% F1. Moreover, **NCC extends beyond text classification, proving effective in multiple-choice QA**, highlighting its adaptability to different NLP tasks. Additionally, **our method shows to be more robust to few-shot example selection and order, requires fewer examples to achieve competitive performance, and the confidences of its predictions are more reliable** than prior methods. By addressing the challenge of multi-token label calibration, NCC enhances the applicability of LLM-based methods to real-world settings where class labels often consist of multiple tokens.

2 Related Work

In-context Learning Several studies have explored various factors influencing ICL’s effectiveness. For instance, [Lu et al. \(2022\)](#) demonstrates that ICL is highly sensitive to the order of in-context examples, while [Liu et al. \(2022\)](#) shows that selecting examples semantically similar to the input can enhance performance. [Min et al. \(2022\)](#) finds that the prompt structure, rather than the specific label values, is a critical factor in classification tasks using ICL. Recent work has shown that ICL is susceptible to *input* length bias, learning to associate labels with inputs based on the input lengths rather than their semantic correctness ([Schoch and Ji, 2025](#)). Complementary to these works, we focus on *label* length biases and their mitigation.

Label Bias Mitigation [Zhao et al. \(2021\)](#) introduces contextual calibration (CC), which uses a content-free input (e.g., “N/A” or the empty string) to estimate the biased distribution of label probabilities. This method assumes that the label probabilities for a context-free input should ideally be uniform. The resulting biases are then used to calibrate the model’s predictions. The authors also categorize biases in ICL into three types: *majority bias* (favoring frequent labels in the prompt), *recency bias* (favoring recently seen examples), and *common token bias* (favoring commonly occurring tokens in the pretraining distribution). However, their approach primarily addresses datasets with few and short tokenized labels, often assuming that class labels are composed of a single token. For multi-token labels, they only consider the first token (e.g., “Ab” for the “Abbreviation” label in the TREC dataset ([Voorhees and Tice, 2000](#))), overlooking the influence of the full label.

Building on this idea, [Fei et al. \(2023\)](#) highlights a new type of label bias (*domain-label bias*) and proposes domain-context calibration to mitigate it. This method adjusts for domain-specific jargon but still relies on single-token labels. Other recent calibration techniques similarly focus on single-token labels ([Han et al., 2023; Jiang et al., 2023b; Zhou et al., 2024](#)). To circumvent multi-token issues, authors modify datasets by reducing labels to single tokens, often avoiding subword tokenization through lowercasing.

While such simplifications may work for some datasets, they are not universally feasible. Datasets like SST-5 ([Socher et al., 2013](#)) contain multi-token labels such as “*very positive*” and “*very negative*”, which become indistinguishable if only the first token (“*very*”) is used. Prior works replace such labels with single-token alternatives (e.g., “*very positive*” with “*great*”, and “*very negative*” with “*terrible*”). However, this approach fails for datasets with many or semantically rich labels (e.g., nuances in labels like “*card arrival*” and “*card delivery estimate*” from the Banking77 dataset ([Casanueva et al., 2020](#)) cannot be adequately represented by a single token). Moreover, representing multi-token labels only by their first token overgeneralizes, as this token may carry probability mass associated with unrelated labels (e.g., “Ab” in TREC could correspond to “*Ability*” or “*About*”).

To address these limitations, [Milios et al. \(2023\)](#) proposes a method in which the model freely generates a label – potentially one that does not exist in the predefined set – and then matches it to the most similar label using a sentence similarity model (e.g., SBERT ([Reimers and Gurevych, 2019](#))). While this avoids the tokenization issue and enables the use of LLMs for text classification with many (and multi-token) labels, it introduces potential errors from the similarity model, which may distort the evaluation of the LLM. Furthermore, calibration cannot be applied to the predictions, as this method relies on free-text generation rather than label probabilities.

Motivated by the absence of a standardized method that fully accounts for multi-token class labels, we first analyze a previously overlooked type of label bias, which we term *label length bias*. To address this, we propose NCC, a label bias mitigation method that explicitly considers the entire label during calibration.

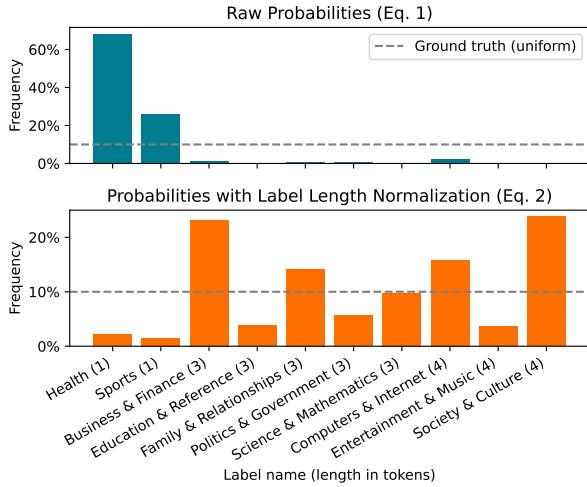


Figure 2: Prediction frequency by label using zero-shot Llama 3.1 (8B) in the Yahoo dataset (Zhang et al., 2015).

3 Label Length Bias in LLMs

LLMs Are Susceptible to Length Bias LLMs are widely used for classification tasks by predicting over a set of candidate labels, either in few-shot (e.g., via ICL) or zero-shot settings. Let \mathcal{L} be the set of possible class labels, where each label $y \in \mathcal{L}$ consists of one or more tokens. Given an input x (and optionally a context C_k of k labeled examples), the model M predicts the label by computing $\text{argmax}_{y \in \mathcal{L}} P_M(y | C_k, x)$. In practice, the probability of a multi-token label y is computed as the product of the conditional probabilities of its tokens (Bengio et al., 2003; Shannon, 1948):

$$P_M(y | C_k, x) = \prod_{i=1}^n P_M(t_i | C_k, x, t_1, \dots, t_{i-1}) \quad (1)$$

where y consists of n tokens t_1, t_2, \dots, t_n . However, this formulation inherently penalizes longer labels, as each additional token multiplies the probability by a number less than one. As a result, longer labels tend to receive lower overall probabilities than shorter ones, regardless of their semantic appropriateness for the input. The top plot in Figure 2 demonstrates this phenomenon: without adjusting for length, the model’s output probabilities are biased toward shorter labels (e.g., “Health” and “Sports”), which are heavily over-predicted. We denote this as *label length bias*, a newly-identified type of bias in LLM-based classification. Previous calibration works (e.g., Zhao et al., 2021; Fei et al., 2023) overlook this type of bias by focusing only on single-token labels and, thus, fail to address it effectively.

Length Normalization Is Not Enough One plausible approach to solve the label length bias in LLMs is to normalize multi-token probabilities by label length (Murray and Chiang, 2018; Brown et al., 2020). By averaging token probabilities, longer labels are no longer penalized purely for their length. However, this introduces a different artifact: in multi-token labels, the probability of the first token is often much lower than that of subsequent tokens, since later tokens are conditioned on the earlier ones and can become highly predictable (e.g., after generating “Business &”, the token “Finance” may have near-certain probability¹). This conditioning inflates the normalized score of frequent or highly predictable label sequences, biasing predictions toward such labels regardless of their overall semantic alignment with the input. As shown in the bottom plot of Figure 2, predictions are now better distributed across label lengths, but this leads to over-prediction of labels with likely token continuations (e.g., “Business & Finance”). Therefore, without further correction, classification is skewed toward token sequences that the model expects to see, rather than labels that truly align with the input context.

4 Normalized Contextual Calibration

In this section, we present NCC, a label bias mitigation method which accounts for label length bias.

When computing the probabilities of multi-token class labels in LLMs, it is essential to address the impact of label length bias. To mitigate it, we propose normalizing the probabilities of labels by their token count. Instead of the raw product of token probabilities, we calculate a normalized probability that accounts for the label length n (Jurafsky and Martin, 2000; Murray and Chiang, 2018):

$$P_M^{\text{norm}}(y | C_k, x) = \sqrt[n]{P_M(y | C_k, x)} \quad (2)$$

This normalization transforms the raw probability into a geometric mean, balancing the contribution of each token and removing the inherent bias against longer labels in LLMs. While there are different ways of normalizing the label length (e.g., average cross-entropy), we use the geometric mean of token probabilities because it allows for a direct application of calibration (where all labels must sum to 1) and is easier to interpret, allowing for confidence reliability analysis (Section 8).

¹Even more so if a single word gets splitted into several tokens (e.g., “Computers” → “Compu” + “ters”).

Dataset	Llama 3.1	Mistral 7B	Qwen2.5	GPT-J
AG News	1.50 _{0.87}	1.50 _{0.87}	1.50 _{0.87}	1.50 _{0.87}
SST-5	1.40 _{0.49}	1.40 _{0.49}	1.40 _{0.49}	1.40 _{0.49}
Yahoo	2.90 _{1.04}	3.00 _{1.18}	2.90 _{1.04}	3.00 _{1.10}
DBpedia	1.64 _{0.72}	1.86 _{1.12}	1.64 _{0.72}	1.93 _{0.88}
20 Newsgroups	2.75 _{1.22}	3.10 _{1.55}	2.75 _{1.22}	2.80 _{1.29}
TREC-50	2.52 _{1.62}	2.80 _{1.67}	2.52 _{1.62}	2.74 _{1.66}
Banking77	3.51 _{1.21}	3.57 _{1.38}	3.51 _{1.21}	3.75 _{1.26}
CLINC150	2.09 _{0.83}	2.15 _{0.95}	2.10 _{0.89}	2.39 _{0.89}

Table 1: Mean and standard deviation of label token lengths for each dataset and model.

Once we have comparable label probabilities unbiased by length, we need to mitigate intrinsic label biases. Following prior works (Zhao et al., 2021; Fei et al., 2023), we use a *content-free* input x_\emptyset – such as an empty string or a neutral placeholder – to probe the model and measure the baseline probability it assigns to each label in the absence of meaningful context. We then calibrate the model’s output probabilities for input x as follows:

$$P_M^{\text{calibrated}}(y | C_k, x) = \frac{P_M^{\text{norm}}(y | C_k, x)}{P_M^{\text{baseline}}(y | C_k, x_\emptyset)} \quad (3)$$

The final predicted label with NCC is obtained by selecting the label with the highest calibrated probability:

$$\hat{y}_{\text{NCC}} = \underset{y \in \mathcal{L}}{\text{argmax}} P_M^{\text{calibrated}}(y | C_k, x) \quad (4)$$

In summary, while normalization addresses biases related to label length, calibration is crucial to ensure that the model’s predictions are driven by the input context, rather than by prior biases toward well-known or frequent label sequences.

5 Experimental Setup

Datasets We evaluate NCC on eight text classification datasets that feature multi-token class labels; cf. Table 1 (and Appendix C for details). These datasets are widely used in prior studies on ICL (Zhao et al., 2021; Min et al., 2022; Fei et al., 2023; Milios et al., 2023; Bertsch et al., 2025). Table 1 confirms that all datasets contain multi-token labels for all models.

Models We evaluate NCC on four LLMs from different families and of different sizes: Llama 3.1 (8B) (Grattafiori et al., 2024), Mistral 7B v0.3 (Jiang et al., 2023a), Qwen2.5 (7B) (Qwen et al., 2025), and GPT-J (6B) (Wang and Komatsuzaki, 2021). These models also differ in how they tokenize class labels (Table 1). In order to be able

to experiment with a large set of models at a reasonable computational cost, we mostly experiment with LLMs with 6B–8B parameters. However, to demonstrate that NCC is also effective for larger models, we additionally evaluate it on the Llama 3.1 70B version (Grattafiori et al., 2024).

Implementation Details Following prior work (Min et al., 2022; Fei et al., 2023), we employ simple and unified prompt templates for all datasets, avoiding task-specific instructions (e.g., explicit label lists) to minimize human engineering. We use $k = 5$ few-shot examples to enable the models to identify task-specific patterns, which are selected randomly, picking an equal number of examples per class as much as possible. Each experiment is repeated five times with different random seeds to account for the sensitivity of ICL to example selection (Liu et al., 2022). Additionally, we evaluate NCC in the zero-shot setting ($k = 0$) to analyze its robustness without task demonstrations. The prompt formats are described in Appendix A. Inference is performed using NVIDIA A100 GPUs.

To calibrate model biases (Eq. 3), we use an ensemble of content-free inputs, as proposed by Zhao et al. (2021). Specifically, we compute the probabilities for each label using five semantically neutral inputs – “” (empty string), “ ” (single space), “N/A”, “[MASK]”, and “Lorem ipsum” – and average them to reduce variance.

Evaluation We report the macro-F1, which accounts for class imbalance (see Table 6). To assess statistical significance, we apply the Wilcoxon signed-rank test (Wilcoxon, 1945).

Baselines We compare NCC against four baselines: (1) standard raw probabilities (Raw Prob; Eq. 1), (2) raw probabilities with length normalization (Norm Prob; Eq. 2), (3) contextual calibration (CC; Zhao et al., 2021),^{2,3} and (4) Gen+SBERT (Milios et al., 2023), a retrieval-based ICL approach that first retrieves the k most similar few-shot examples, then lets the LLM freely generate a label (not restricted to the label space), and finally

²We adopt CC as a representative standard calibration baseline due to its simplicity and wide recognition in the literature. For completeness, we compare NCC against other relevant calibration techniques in Appendix D.3.

³For CC, we extend the original single-token implementation in Zhao et al. (2021) with Eq. 3, where the baseline probabilities are obtained from Eq. 1 – with the single-token implementation, labels “very positive” and “very negative” would be treated as the same token (“very”).

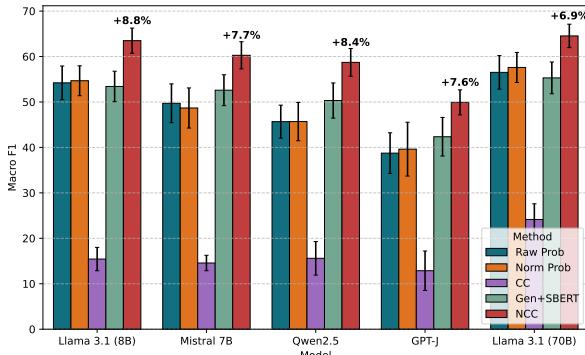


Figure 3: Few-shot performance of methods and models, averaged over all datasets. Error bars indicate the standard deviation across runs. Numbers above the bars indicate absolute improvement of NCC over the second-best method.

matches the generated label to the closest candidate using a sentence similarity model (e.g., SBERT).

6 Results

Few-shot Results Table 2 reports detailed results across models, datasets, and methods with $k = 5$ examples, while Figure 3 summarizes overall trends. NCC consistently outperforms all baselines with statistically significant improvements, achieving absolute macro-F1 gains ranging from +6.9% to +8.8% over the second-best method (7.6% on average). The gains are most pronounced on datasets with longer and more complex labels, such as Banking77 and CLINC150, where mitigating label length bias is most critical.

NCC also surpasses retrieval-based ICL approaches such as Gen+SBERT, which leverages *similar* few-shot examples and free-form label generation followed by similarity matching to candidate labels. Despite using *randomly* selected examples, NCC achieves higher performance, suggesting that properly calibrated label probabilities are more beneficial than retrieval similarity alone – a common approach in practical applications.

Two model-specific trends emerge: (1) NCC consistently improves performance across all model sizes, including Llama 3.1 70B, which shows that even larger LLMs suffer from label bias and require calibration; (2) NCC narrows the performance gap between large and small models, indicating that proper calibration can make smaller models competitive (see Appendix D.1 for details).

Notably, the standard CC method for multi-token labels (adapted from Zhao et al., 2021) achieves near-zero F1 on some datasets (e.g., TREC-50).

Without length normalization, this method inherits the bias toward shorter labels (higher P_{baseline}). During calibration (Eq. 3), this results in excessive penalization of short labels and overcompensation for longer ones, skewing predictions toward lengthy labels regardless of input relevance (details in Appendix E). This illustrates the necessity of normalizing multi-token probabilities to prevent calibration from amplifying label length bias.

NCC underperforms the standard raw probability on AG News and SST-5, where the full label set (4 and 5 classes, respectively) is already represented in the few-shot examples ($k = 5$) and label semantics are simple. In such cases, NCC introduces unnecessary adjustments that degrade performance. This aligns with our zero-shot findings (below), where NCC becomes essential to mitigate label bias when the label space is not fully represented.

Zero-shot Results We further evaluate NCC in a zero-shot setting, where no demonstrations are provided. As shown in Figure 4, NCC maintains its advantage over all baselines across most datasets and models, mirroring trends from the few-shot scenario (detailed results in Appendix D.2).

While all methods yield lower performance due to the lack of examples as expected, NCC shows the smallest relative performance difference between zero- and few-shot settings (36.8%; Figure 5), indicating that it is less dependent on labeled examples. In some cases, NCC’s *zero-shot* performance even exceeds the *few-shot* raw baseline (e.g., GPT-J average F1: 42.4 vs. 38.7). This further demonstrates the effectiveness of NCC’s label bias mitigation, as it allows models to leverage the unbiased label semantics and perform better on zero-shot than on standard few-shot learning from examples.

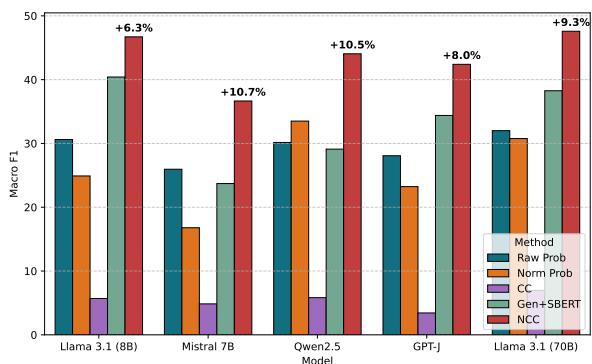


Figure 4: Zero-shot performance of methods and models, averaged over all datasets. Numbers indicate absolute improvement of NCC over the second-best method.

Method	AG News	SST-5	Yahoo	DBpedia	20 Newsgroups	TREC-50	Banking77	CLINC150	Avg.
<i>Llama 3.1 (8B)</i>									
Raw Prob	83.8 _{6.2}	42.3_{9.1}*	61.3 _{2.7}	83.4 _{0.7}	41.0 _{3.4}	28.2 _{1.9}	37.8 _{4.6}	55.9 _{1.2}	54.2 _{3.7}
Norm Prob	85.3_{1.8}*	38.6 _{6.3}	60.6 _{2.2}	80.0 _{4.7}	43.3 _{3.2}	29.3 _{2.4}	36.3 _{3.5}	64.1 _{2.2}	54.7 _{3.3}
CC	84.0 _{2.3}	23.9 _{7.4}	8.1 _{3.5}	1.0 _{0.0}	5.3 _{6.1}	0.2 _{0.0}	0.1 _{0.0}	1.0 _{1.1}	15.4 _{2.6}
Gen+SBERT	80.4 _{5.1}	41.2 _{5.3}	64.3 _{1.1}	67.3 _{3.9}	40.5 _{2.1}	33.8_{2.9}*	47.5 _{3.1}	52.3 _{3.2}	53.4 _{3.3}
NCC	78.0 _{5.0}	38.2 _{8.1}	65.1_{3.5}	92.5_{0.7}*	67.7_{0.9}*	33.8_{2.1}	59.5_{0.9}*	73.1_{1.0}*	63.5_{2.8}*
<i>Mistral 7B</i>									
Raw Prob	81.2 _{6.4}	40.7 _{7.7}	52.9 _{3.2}	67.7 _{5.8}	39.1 _{4.1}	28.6 _{1.4}	34.5 _{3.9}	53.1 _{1.6}	49.7 _{4.3}
Norm Prob	83.1_{2.8}*	33.4 _{6.1}	56.2 _{3.2}	62.6 _{3.1}	32.5 _{7.1}	26.8 _{5.3}	32.9 _{5.0}	61.9 _{2.6}	48.7 _{4.4}
CC	81.5 _{1.7}	21.1 _{3.8}	7.4 _{3.7}	1.0 _{0.0}	4.0 _{3.5}	0.2 _{0.0}	0.1 _{0.0}	1.4 _{0.9}	14.6 _{1.7}
Gen+SBERT	81.9 _{2.3}	40.9_{7.3}*	64.3_{1.4}	62.0 _{3.3}	39.2 _{3.1}	36.4_{5.1}	46.9 _{1.5}	49.1 _{3.1}	52.6 _{3.4}
NCC	76.8 _{4.1}	34.7 _{7.3}	62.1 _{2.0}	91.2_{1.6}*	60.4_{4.2}*	34.2 _{1.8}	57.9_{1.1}*	64.8_{1.9}*	60.3_{3.0}*
<i>Qwen2.5 (7B)</i>									
Raw Prob	74.6 _{4.7}	35.8 _{4.6}	47.1 _{1.5}	74.1 _{3.7}	29.4 _{4.7}	25.1 _{2.4}	23.6 _{5.5}	55.6 _{2.0}	45.7 _{3.6}
Norm Prob	83.1_{2.4}*	42.5_{4.3}*	44.6 _{2.2}	72.8 _{4.2}	21.3 _{7.5}	19.1 _{5.2}	18.3 _{4.6}	63.8 _{3.4}	45.7 _{4.2}
CC	49.2 _{5.9}	24.0 _{5.9}	35.4 _{11.4}	3.1 _{1.3}	10.4 _{3.6}	0.2 _{0.0}	0.1 _{0.0}	2.7 _{1.4}	15.6 _{3.7}
Gen+SBERT	74.6 _{3.3}	37.6 _{4.8}	59.1_{1.8}	71.1 _{4.2}	30.8 _{4.3}	33.8 _{5.1}	41.1 _{4.1}	54.5 _{3.4}	50.3 _{3.9}
NCC	71.7 _{3.8}	35.5 _{5.8}	59.1_{2.5}	87.1_{1.9}*	58.3_{2.8}*	34.6_{2.9}	52.3_{2.4}*	71.1_{2.4}*	58.7_{3.1}*
<i>GPT-J (6B)</i>									
Raw Prob	67.3 _{1.5}	25.7_{3.9}	30.6 _{12.5}	70.8 _{3.1}	17.2 _{9.6}	24.0 _{1.7}	29.8 _{2.7}	44.5 _{0.9}	38.7 _{4.5}
Norm Prob	74.8_{3.2}*	25.4 _{4.6}	43.5 _{14.8}	65.0 _{6.2}	10.3 _{9.4}	14.5 _{5.3}	32.4 _{2.3}	51.0 _{1.5}	39.6 _{5.9}
CC	70.7 _{18.0}	12.5 _{6.9}	12.4 _{4.8}	1.0 _{0.0}	4.7 _{4.5}	0.2 _{0.0}	0.1 _{0.0}	1.4 _{0.6}	12.9 _{4.3}
Gen+SBERT	66.0 _{0.7}	25.0 _{2.0}	46.4 _{14.8}	63.5 _{1.5}	18.0 _{9.4}	28.2_{3.0}*	43.1 _{1.0}	48.6 _{1.7}	42.3 _{4.3}
NCC	67.7 _{1.5}	24.4 _{3.5}	46.5_{5.8}	90.6_{0.8}*	47.8_{4.6}*	21.7 _{1.1}	43.2_{1.6}	57.4_{3.1}*	49.9_{2.8}*
<i>Llama 3.1 (70B)</i>									
Raw Prob	88.8_{2.0}*	48.0 _{8.4}	69.7 _{1.3}	80.8 _{3.1}	43.2 _{4.4}	29.7 _{1.2}	36.2 _{6.6}	55.8 _{2.5}	56.5 _{3.7}
Norm Prob	87.9 _{1.0}	48.1_{6.2}*	70.4_{0.4}*	80.1 _{2.8}	50.3 _{2.9}	26.1 _{5.1}	31.0 _{6.1}	66.9 _{1.9}	57.6 _{3.3}
CC	86.5 _{1.3}	27.4 _{6.8}	53.1 _{6.1}	21.5 _{10.7}	0.4 _{0.0}	0.2 _{0.0}	0.8 _{0.7}	3.3 _{2.1}	24.1 _{3.4}
Gen+SBERT	84.3 _{2.5}	47.0 _{3.4}	62.9 _{2.5}	62.5 _{3.9}	47.3 _{4.7}	37.1_{3.2}*	45.9 _{5.0}	55.3 _{2.8}	55.3 _{3.5}
NCC	84.6 _{4.8}	37.8 _{3.9}	65.0 _{1.3}	98.2_{0.3}	73.5_{1.5}*	32.0 _{2.8}	56.6_{3.3}*	68.5_{2.6}*	64.5_{2.6}*

Table 2: Few-shot performance (macro-F1_{±std}) of the conventional approach (Raw Prob; Brown et al. 2020), conventional approach with label length normalization (Norm Prob), contextual calibration (CC; Zhao et al. 2021), free generation plus similarity matching (Gen+SBERT; Milioto et al. 2023), and normalized contextual calibration (NCC; *Ours*). * indicates significant dominance of (or over) NCC at p < 0.05.

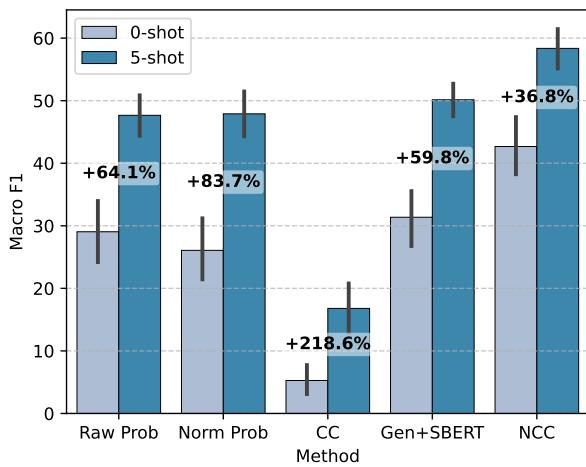


Figure 5: Average performance of all methods in the zero-shot and few-shot setting. Numbers indicate relative improvement when going from 0 to 5 shots.

Our findings align with those in Min et al. (2022), who emphasize that defining the label space is crucial for ICL performance. In the few-shot experiments where all possible labels are included in the context (e.g., AG News and SST-5 in Table 2), the model is already well-calibrated by the explicit definition of the label space, so NCC introduces unnecessary adjustments that degrade performance. In contrast, in the zero-shot setting, when the label space is only implicitly defined, NCC becomes essential to correct the model’s inherent biases and improve performance. This highlights NCC’s role as a necessary calibration step when the label space is incomplete or underspecified in the prompt.

7 Generalizability of NCC to Other Tasks

Another key contribution of NCC is its ability to extend the applicability of calibration techniques beyond classification tasks. Prior calibration methods (e.g., Zhao et al. (2021)) are limited to scenarios with single-token labels. By extending calibration with label length bias mitigation, NCC enables bias correction in previously inaccessible tasks.

We demonstrate this capability through multiple-choice question answering (MCQA), a task that requires selecting the correct option for a question among a set of choices. For this experiment, we adopt a cloze prompt formulation (Brown et al., 2020), a widely used MCQA approach that involves getting the model probabilities for each independent completion (choice), without showing the list of options to the model (detailed prompt can be found in Appendix A). Thus, the model does not know the label distribution⁴ and might exhibit label biases. Table 3 evaluates NCC on three diverse MCQA datasets. The results highlight the effectiveness of NCC in these tasks, achieving the highest average performance across all models.

Unlike text classification, MCQA does not exhibit *majority* or *recency* biases (Zhao et al., 2021), as each example has its own unique set of candidate answers. Instead, the primary challenge lies in addressing *common token* and *label length* biases, which can distort the model’s probability estimates. While gains are smaller than in text classification – likely due to the absence of *majority* and *recency* biases, diminishing the effect of calibration – these consistent improvements demonstrate the potential of NCC to generalize beyond text classification. By enabling calibration techniques to handle multi-token labels (or options) effectively, NCC opens the door to applying these methods in a wide array of tasks that require mitigation of label biases.

8 Analysis

NCC Reduces Sensitivity to Example Selection

Prior work has shown that ICL is highly sensitive to the choice and order of few-shot examples (Liu et al., 2022; Lu et al., 2022). To assess whether NCC reduces this sensitivity, we compare the average standard deviation and coefficient of variation of F1 scores across different random seeds for each model–dataset combination. As shown in Table 4, NCC exhibits the lowest variability,

Method	OBQA	CSQA	QASC	Avg.
<i>Llama 3.1 (8B)</i>				
Raw Prob	24.8 _{0.5}	55.7 _{1.4}	51.3 _{4.4}	43.9 _{2.1}
Norm Prob	32.8 _{1.0}	57.3 _{1.2}	51.6 _{4.1}	47.2 _{2.1}
CC	33.2 _{1.2}	34.7 _{0.8}	22.8 _{2.0}	30.2 _{1.3}
Gen+SBERT	29.8 _{1.2}	45.7 _{0.5}	43.9 _{1.6}	39.8 _{1.1}
NCC	37.9_{1.0}	58.4_{0.8}	59.3_{2.9}	51.9_{1.6}
<i>Mistral 7B</i>				
Raw Prob	23.2 _{1.2}	57.5 _{0.9}	52.8 _{2.8}	44.5 _{1.6}
Norm Prob	33.7_{1.4}	56.9 _{1.4}	52.2 _{2.0}	47.6 _{1.6}
CC	33.3 _{1.5}	31.1 _{0.9}	18.0 _{1.2}	27.5 _{1.2}
Gen+SBERT	29.3 _{1.0}	47.8 _{1.0}	44.3 _{0.4}	40.5 _{0.8}
NCC	32.1 _{1.1}	57.6_{1.2}	57.4_{1.9}	49.0_{1.4}
<i>Qwen2.5 (7B)</i>				
Raw Prob	25.7 _{1.0}	59.5 _{0.7}	54.5 _{3.3}	46.6 _{1.7}
Norm Prob	33.2 _{0.5}	60.5_{0.4}	53.0 _{1.9}	48.9 _{0.9}
CC	33.6 _{0.8}	36.4 _{1.3}	20.4 _{0.9}	30.2 _{1.0}
Gen+SBERT	31.4 _{1.3}	52.5 _{0.4}	46.5 _{0.7}	43.4 _{0.8}
NCC	39.5_{0.9}	57.7 _{1.2}	62.6_{1.5}	53.3_{1.2}
<i>GPT-J (6B)</i>				
Raw Prob	17.0 _{1.1}	45.3 _{2.0}	37.8 _{2.0}	33.4 _{1.7}
Norm Prob	23.0 _{1.2}	44.5 _{1.3}	38.3 _{1.4}	35.3 _{1.3}
CC	30.3_{0.9}	24.8 _{0.7}	13.4 _{0.4}	22.8 _{0.7}
Gen+SBERT	23.9 _{0.6}	38.2 _{1.4}	38.0 _{1.2}	33.3 _{1.1}
NCC	26.3 _{1.5}	46.9_{0.6}	43.5_{1.4}	38.9_{1.2}
<i>Llama 3.1 (70B)</i>				
Raw Prob	29.1 _{0.5}	61.6 _{1.8}	60.0 _{1.6}	50.2 _{1.3}
Norm Prob	37.4 _{1.1}	63.5 _{1.9}	58.4 _{1.1}	53.1 _{1.4}
CC	36.5 _{0.7}	37.2 _{2.0}	25.4 _{0.5}	33.0 _{1.1}
Gen+SBERT	33.1 _{1.8}	51.2 _{1.0}	49.9 _{0.9}	44.8 _{1.2}
NCC	44.4_{1.7}	64.1_{1.8}	68.1_{0.7}	58.8_{1.4}

Table 3: Performance (Macro F1) for the multiple choice QA task. Dataset abbreviations: OBQA (Open-BookQA; Mihaylov et al. 2018), CSQA (CommonsenseQA; Talmor et al. 2019), QASC (Question Answering in Context; Khot et al. 2020).

indicating greater robustness and stability across different example sets.

Calibration Level vs. Few-shot Examples We analyze how the number of few-shot examples (k) affects calibration by comparing NCC and the standard Raw Prob baseline across varying k ’s (equal to each dataset’s class count) on all datasets. We select examples randomly from the train set, smaller sets of examples are always subsets of larger sets, and we maintain a fixed order to avoid permutation sensitivity (Lu et al., 2022). We measure the required calibration level using the Kullback-Leibler (KL) divergence between the raw and the calibrated (NCC) probability distributions, reflecting the divergence between them. Due to computational cost,

⁴In MCQA, the “labels” are the different choices.

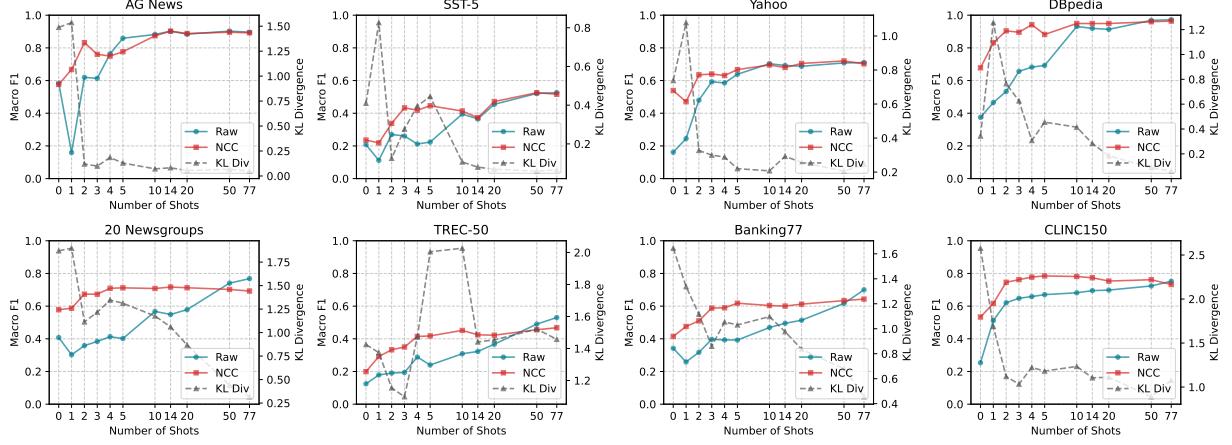


Figure 6: Performance and calibration level for different k values, run with Llama 3.1 (8B).

Method	Std. Dev.	Coef. Var.
Raw Prob	0.039	0.103
Norm Prob	0.042	0.134
CC	0.034	0.303
Gen+SBERT	0.038	0.088
NCC	0.031	0.058

Table 4: Average sensitivity (standard deviation and coefficient of variation) of F1 scores across different random example selections.

we use Llama 3.1 (8B) with a single random seed.

Figure 6 shows that NCC generally outperforms Raw Prob, with the performance gap narrowing as k increases. This trend is expected because fewer examples require higher calibration to correct biases, as the model has fewer data to learn the label space. The decreasing KL divergence with increasing k confirms this, indicating that raw predictions become more aligned with calibrated probabilities. This suggests NCC is particularly effective in scenarios with limited few-shot examples, where calibration is crucial for mitigating label biases. Additionally, in settings with a large number of classes (bottom row of Figure 6), very few examples (2–5) in NCC perform almost as good as the many-examples (50–77) raw ICL counterparts. This has significant implications for practical applications, as the number of examples can be drastically reduced while keeping comparable performance.

NCC Enhances Reliability of Predictions Another crucial aspect when evaluating LLMs is considering how reliable its predictions are. This is measured by the alignment between model confidence (probability) and prediction accuracy (Guo et al., 2017). Figure 7 shows the reliability dia-

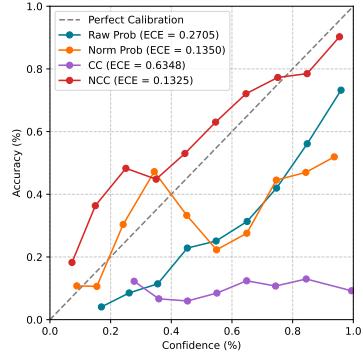


Figure 7: Reliability diagrams for all methods. The x-axis are bins of predicted confidence, and the y-axis shows the accuracy for the given bin.

grams for the different methods, averaged across all datasets for the Llama 3.1 (8B) model. NCC is the most reliable method, showing the closest line to the perfect calibration and the lowest expected calibration error (ECE). This result demonstrates that NCC not only improves overall model performance, but also the reliability of the predictions. (Further analysis is conducted in Appendix E.)

9 Conclusion

In this work, we highlight the previously overlooked issue of multi-token label length bias in LLMs and introduce NCC as an effective solution. By normalizing probabilities and applying calibration with prior probabilities at the full-label level, NCC significantly improves performance over existing methods across diverse datasets and models. Our approach extends bias mitigation to settings such as multiple-choice question answering, demonstrating its versatility beyond classification tasks. Additionally, NCC reduces sensitivity to few-shot example selection, requires fewer exam-

ples to perform competitively, and produces more reliable confidence estimates. Our findings underscore the importance of addressing full-label biases to enhance the performance and reliability of LLM-based systems, especially in real-world scenarios where class labels often consist of multiple tokens.

Limitations

While NCC demonstrates strong performance and generalizability across various tasks, several limitations should be considered.

First, the applicability of NCC is constrained by the requirement to access token-level probabilities for all possible class labels. This restricts its use to open-source models that can be run locally, as many proprietary or cloud-based models do not provide full token probability distributions. Some APIs (e.g., OpenAI’s `logprobs` parameter) offer token probabilities but are typically limited to the top-20 tokens, making NCC infeasible for tasks requiring complete probability distributions.

Second, NCC is not suitable for open-ended tasks where the label space is not predefined. Since the method relies on calibrating prior biases across a fixed set of labels, it cannot be directly applied to tasks with an unrestricted output space, such as open-ended text generation.

Third, NCC is not designed for full-sentence generation tasks. Calibration is most effective when applied to tasks where the label itself carries minimal semantic meaning (e.g., class labels in classification, or sentence continuations in multiple-choice question answering). In contrast, full-sentence responses inherently carry substantial baseline probability due to their coherence and fluency, regardless of calibration. Applying NCC to meaningful text generation would likely penalize fluent responses rather than correct systematic biases.

Despite these limitations, NCC remains a valuable tool for mitigating label bias and improving LLM performance in settings where class labels are well-defined.

Ethics Statement

Our work demonstrates that NCC significantly improves LLM performance for text classification and related NLP tasks by mitigating label biases. However, these improvements do not fully eliminate inherent risks, and LLMs remain susceptible to errors. Therefore, we caution against relying solely on LLMs in critical settings, such as medical

advice, without appropriate human oversight and domain-specific validation.

Acknowledgments

This work was supported by the Carl Zeiss Foundation through the MAINCE project (grant number P2022-08-009).

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Amanda Bertsch, Maor Ivgi, Emily Xiao, Uri Alon, Jonathan Berant, Matthew R. Gormley, and Graham Neubig. 2025. In-context learning with long-context models: An in-depth exploration. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12119–12149, Albuquerque, New Mexico. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. 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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 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*.

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](#). *Preprint*, arXiv:2310.06825.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. 2023b. [Generative calibration for in-context learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2312–2333, Singapore. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. [Qasc: A dataset for question answering via sentence composition](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Ken Lang. 1995. Newsweeder: learning to filter news. In *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, ICML’95, page 331–339, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. [An evaluation dataset for intent classification and out-of-scope prediction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

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.

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.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. [Can a suit of armor conduct electricity? a new dataset for open book question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. 2023. [In-context learning for text classification with many labels](#). In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 173–184, Singapore. Association for Computational Linguistics.

Sewon Min, Xinxin Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. [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.

Kenton Murray and David Chiang. 2018. [Correcting length bias in neural machine translation](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.

Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Hossain Shaikh Saadi, Minh Duc Bui, Mario Sanz-Guerrero, and Katharina Von Der Wense. 2025. [JGU Mainz’s submission to the WMT25 shared task on LLMs with limited resources for Slavic languages: MT and QA](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1151–1157, Suzhou, China. Association for Computational Linguistics.

Mario Sanz-Guerrero, Minh Duc Bui, and Katharina von der Wense. 2025. [Mind the gap: A closer look at](#)

tokenization for multiple-choice question answering with LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19584–19594, Suzhou, China. Association for Computational Linguistics.

Mario Sanz-Guerrero and Katharina von der Wense. 2025. *Corrective in-context learning: Evaluating self-correction in large language models*. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 24–33, Albuquerque, New Mexico. Association for Computational Linguistics.

Stephanie Schoch and Yangfeng Ji. 2025. *In-context learning (and unlearning) of length biases*. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7633–7671, Albuquerque, New Mexico. Association for Computational Linguistics.

Sagi Shaier, Mario Sanz-Guerrero, and Katharina von der Wense. 2025. *Asking again and again: Exploring LLM robustness to repeated questions*. *Preprint*, arXiv:2412.07923.

C. E. Shannon. 1948. *A mathematical theory of communication*. *The Bell System Technical Journal*, 27(3):379–423.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. *Recursive deep models for semantic compositionality over a sentiment treebank*. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. *CommonsenseQA: A question answering challenge targeting commonsense knowledge*. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Ellen M. Voorhees and Dawn M. Tice. 2000. *Building a question answering test collection*. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00*, page 200–207, New York, NY, USA. Association for Computing Machinery.

Ben Wang and Aran Komatsuzaki. 2021. *GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model*. <https://github.com/kingoflolz/mesh-transformer-jax>.

Frank Wilcoxon. 1945. *Individual comparisons by ranking methods*. *Biometrics Bulletin*, 1(6):80–83.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. *Character-level convolutional networks for text classification*. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. *Calibrate before use: Improving few-shot performance of language models*. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. 2024. *Batch calibration: Rethinking calibration for in-context learning and prompt engineering*. In *The Twelfth International Conference on Learning Representations*.

A Prompt Formats

For the prompt formats, we follow previous work and employ simple and unified prompt templates for both text classification and question answering tasks (Min et al., 2022; Fei et al., 2023; Sanz-Guerrero and von der Wense, 2025; Shaier et al., 2025; Saadi et al., 2025).

The prompts end with “Label:” (for text classification tasks; Figure 8) and “Answer:” (for question answering tasks; Figure 9), and we extract the model’s output probabilities for each candidate label/answer starting from that position. Following Sanz-Guerrero et al. (2025), the first token of each label/answer is prefixed with a space to align with the model’s default tokenization, as this has been shown to improve performance and model calibration.

Text Classification Prompt	
Text:	{example_1}
Label:	{ground_truth_1}
Text:	{example_2}
Label:	{ground_truth_2}
...	
Text:	{example_k}
Label:	{ground_truth_k}
Text:	{input_text}
Label:	

Figure 8: Prompt format for text classification tasks.

B Illustration of the Importance of NCC

To illustrate the necessity of NCC for handling multi-token labels, we revisit the toy example il-

Question Answering Prompt				
Question: {example_1}				
Answer: {ground_truth_1}				
Question: {example_2}				
Answer: {ground_truth_2}				
...				
Question: {example_k}				
Answer: {ground_truth_k}				
Question: {input_text}				
Answer:				

Figure 9: Prompt format for question answering tasks.

lustrated in Figure 1. Consider a sentiment classification task similar to the SST-5 dataset, where the five class labels are: “*very positive*”, “*positive*”, “*neutral*”, “*negative*”, and “*very negative*”. To highlight the impact of calibration, we introduce a sixth, naive class label verbalized as:

“It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of light, it was the season of darkness, it was the spring of hope, it was the winter of despair.”

taken from “A Tale of Two Cities” by Charles Dickens. This label is intentionally much longer than the other labels and consists of a well-known sequence of words, making it a strong candidate for unintended biases.

For simplicity, we use the Llama 3.1 (8B) model with a simple few-shot prompt:

Text: This is a good product, and I'd probably recommend it to others.
 Label: positive

Text: The presentation was informative but not particularly engaging.
 Label: neutral

Text: The service was slow, and the food was cold.
 Label: negative

Text: This product has completely changed my life for the better!
 Label:

The model must predict the sentiment of the last sentence, whose ground-truth label is “*very positive*”. Table 5 presents the probabilities assigned to each class by the different methods.

The standard raw probability approach assigns a reasonable probability distribution but incorrectly

Class Label	Raw	Norm	CC	NCC
very positive	0.13	0.20	1e-9	0.81
positive	0.86	0.39	2e-12	0.14
neutral	2e-3	1e-3	7e-15	4e-4
negative	3e-3	1e-3	3e-14	1e-3
very negative	5e-5	4e-3	8e-12	9e-3
<i>it was the best of times, [...]</i>	1e-14	0.41	0.99	0.04

Table 5: Probabilities assigned to each class in the toy example. **Bold** values indicate the highest probability (i.e., the predicted class).

predicts “*positive*” instead of “*very positive*”. This happens because “*very positive*” consists of two tokens (“*very*” and “*positive*”), meaning its probability is the product of individual token probabilities, making it disproportionately lower.

Length normalization (Norm Prob) mitigates this length bias by normalizing multi-token probabilities. However, it assigns a high probability to the naive class label due to the high compound probability of its familiar token sequence, despite its irrelevance to the task.

CC (contextual calibration; Zhao et al. 2021) attempts to correct label biases but fails in this case. Since the standard raw method assigns a very low prior probability to the naive (longest) label, CC overcompensates and amplifies its probability, leading to incorrect predictions.

NCC (normalized contextual calibration; *ours*) successfully mitigates both length bias and overcompensation. By normalizing probabilities before calibration, it ensures that class probabilities are fairly adjusted, leading to a more accurate prediction of “*very positive*”.

This example highlights the importance of properly handling full-label probabilities, particularly when dealing with multi-token labels, to ensure reliable and unbiased predictions of LLMs.

C Dataset Details

We evaluate NCC using eight text classification datasets and three multiple-choice QA datasets, spanning diverse domains and label complexities. Table 6 summarizes key characteristics including class count, balance status, test set size, and example labels.

C.1 20 Newsgroups Dataset

While widely adopted in traditional NLP research, the 20 Newsgroups dataset (Lang, 1995) has been largely overlooked in recent LLM-based classification studies due to its non-descriptive class labels

Dataset	# Classes	Balanced	Test	Class Labels
<i>Text Classification</i>				
AG News (Zhang et al., 2015)	4	✓	500	“World”, “Sports”, “Business”, “Science and Technology”
SST-5 (Socher et al., 2013)	5	✗	1,100	“very positive”, “positive”, “neutral”, “negative”, “very negative”
Yahoo (Zhang et al., 2015)	10	✓	500	Topics like “Society”, “Science”, “Health”, “Education”, and others.
DBpedia (Zhang et al., 2015)	14	✓	500	Types of entities, such as “Company”, “Educational Institution”, “Artist”, and more.
20 Newsgroups (Lang, 1995)	20	✗	500	Categories are shown in Table 7.
TREC-50 (Voorhees and Tice, 2000)	50	✗	500	Fine-grained question types like “Invention, book and other creative piece”, “Description of a person”, and “Distance, linear measure”.
Banking77 (Casanueva et al., 2020)	77	✗	924	Intents such as “card activation”, “transaction failed”, “account blocked”, and more.
CLINC150 (Larson et al., 2019)	150	✗	3,000	Intents in domains like banking, travel, and work, such as “transfer money”, “book flight”, and “schedule meeting”.
<i>Multiple Choice QA</i>				
OpenBookQA (Mihaylov et al., 2018)	4	N/A	500	N/A
CommonsenseQA (Talmor et al., 2019)	5	N/A	1,221	N/A
QASC (Khot et al., 2020)	8	N/A	926	N/A

Table 6: Full dataset information.

(e.g., “*sci.med*”). In traditional methods, such as bag-of-words models, class label wording was irrelevant since these approaches relied solely on feature representations of the input text. However, LLMs benefit from more representative and meaningful labels, as they utilize the wording of class labels during inference. To address this, we remap the original categories as shown in Table 7 to enhance interpretability and facilitate a more effective use of this dataset with LLMs.

Original Class Label	New Class Label
alt.atheism	Atheism & Secularism
comp.graphics	Computer Graphics
comp.os.ms-windows.misc	Windows OS
comp.sys.ibm.pc.hardware	IBM PC Hardware
comp.sys.mac.hardware	Macintosh Hardware
comp.windows.x	X Window System
misc.forsale	Classified Ads
rec.autos	Automobiles
rec.motorcycles	Motorcycles
rec.sport.baseball	Baseball
rec.sport.hockey	Hockey
sci.crypt	Cryptography
sci.electronics	Electronics
sci.med	Medicine
sci.space	Space & Astronomy
soc.religion.christian	Christianity
talk.politics.guns	Gun Politics
talk.politics.mideast	Middle East Politics
talk.politics.misc	General Politics
talk.religion.misc	General Religion

Table 7: Mapping of the original 20 Newsgroups class labels to more representative and descriptive labels for use with LLMs.

D Detailed Results

D.1 Model Size Comparison

In the main text, we demonstrate the effectiveness of NCC across different LLMs of different sizes (Section 6). To further evaluate the impact of model size on NCC performance, we stick with the Llama 3.1 model and compare its performance with different sizes (8B, 70B) on the text classification task.

Figure 10 shows the average performance and relative improvement from the 8B version to the 70B version of Llama 3.1 for each method under the same settings. NCC enables the smaller model to achieve the most similar performance to the larger model, with only a 1.6% gap in average F1, despite being almost nine times smaller. This is a significant drop in performance gap, compared to the 4.3% difference across model sizes in the standard raw probability approach. This result highlights the effectiveness of NCC in enhancing the performance of smaller models, enabling them to achieve comparable results to larger models in text classification when properly calibrated.

D.2 Zero-Shot Results

Table 8 provides the full results for all methods on each dataset and model in the zero-shot setting (corresponding to Figure 4 in the main text).

Method	AG News	SST-5	Yahoo	DBpedia	20 Newsgroups	TREC-50	Banking77	CLINC150	Avg.
<i>Llama 3.1 (8B)</i>									
Raw Prob	58.5	20.6	16.2	36.8	40.8	12.5	34.2	25.4	30.6
Norm Prob	17.7	14.0	36.0	57.6	18.7	3.5	26.5	25.4	24.9
CC	10.0	16.4	11.6	1.0	5.7	0.2	0.0	0.7	5.7
Gen+SBERT	56.2	24.0	45.9	44.7	29.1	20.2	47.2	55.9	40.4
NCC	57.8	23.3	52.6	67.4	57.8	19.9	41.5	53.3	46.7
<i>Mistral 7B</i>									
Raw Prob	54.8	14.0	10.3	30.2	39.5	12.9	31.6	14.5	26.0
Norm Prob	31.6	10.8	13.3	5.2	23.1	5.6	27.8	16.8	16.8
CC	10.0	15.8	11.0	1.0	0.4	0.2	0.0	0.5	4.9
Gen+SBERT	43.5	21.9	26.3	34.7	6.6	13.5	25.2	18.1	23.7
NCC	48.9	26.4	41.9	44.0	48.9	15.9	30.5	36.6	36.7
<i>Qwen2.5 (7B)</i>									
Raw Prob	54.3	22.7	18.3	49.2	32.9	20.3	29.0	14.5	30.2
Norm Prob	46.5	31.1	34.0	65.9	22.2	7.6	32.1	28.5	33.5
CC	10.0	12.9	20.7	1.0	1.3	0.2	0.0	0.6	5.8
Gen+SBERT	38.0	16.2	31.6	38.8	19.5	19.2	36.4	33.1	29.1
NCC	57.7	23.5	45.3	69.1	50.3	29.2	36.5	40.8	44.1
<i>GPT-J (6B)</i>									
Raw Prob	53.3	16.9	8.3	42.4	34.1	15.8	27.8	26.1	28.1
Norm Prob	26.2	13.8	14.2	52.2	17.2	15.2	24.1	22.8	23.2
CC	10.0	10.6	1.8	1.0	3.0	0.2	0.0	0.9	3.4
Gen+SBERT	53.3	21.3	38.7	47.5	18.7	19.1	37.9	38.5	34.4
NCC	59.8	21.0	47.0	52.8	45.8	25.0	37.4	50.4	42.4
<i>Llama 3.1 (70B)</i>									
Raw Prob	57.5	22.2	15.9	40.3	43.1	14.9	33.5	28.7	32.0
Norm Prob	22.1	30.2	37.4	56.4	26.0	13.4	22.9	37.8	30.8
CC	10.0	16.2	22.2	1.0	5.7	0.2	0.0	0.6	7.0
Gen+SBERT	54.8	24.0	44.5	45.1	28.4	18.5	39.1	51.6	38.3
NCC	64.3	32.5	51.9	62.2	58.0	20.5	38.0	53.3	47.6

Table 8: Performance comparisons (macro-F1) for the zero-shot setting. There is no \pm std because there are no examples, resulting in consistent performance across runs.

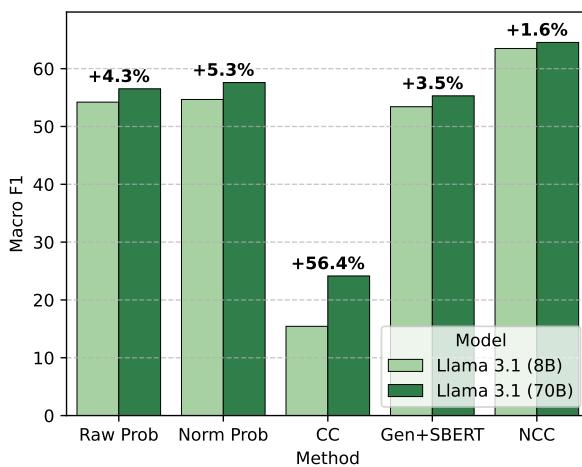


Figure 10: Average performance of all methods with Llama 3.1 (8B) and Llama 3.1 (70B). Numbers indicate relative improvements when scaling from 8B to 70B.

D.3 Comparison with Other Calibration Methods

In the main text, we compare NCC with contextual calibration (CC; Zhao et al. 2021) as a representative standard calibration method, due to its simplicity and wide recognition in the community. However, there are several other relevant calibration methods proposed in recent literature.

Here, we extend the comparison to include domain-context calibration (DC; Fei et al. 2023), generative calibration (GC; Jiang et al. 2023b), and batch calibration (BC; Zhou et al. 2024). For these methods, we follow the original papers’ implementations (e.g., using random contexts from the training set for DC, and $L = 100$ generated inputs from the model for GC to estimate priors). Moreover, we apply our normalized calibration approach to these methods, resulting in normalized domain-

Method	AG News	SST-5	Yahoo	DBpedia	20 Newsgroups	TREC-50	Banking77	CLINC150	Avg.
<i>No Calibration</i>									
Raw Prob	83.8	42.3	61.3	83.4	41.0	28.2	37.8	55.9	54.2
Norm Prob	85.3	38.6	60.6	80.0	43.3	29.3	36.3	64.1	54.7
Gen+SBERT	80.4	41.2	64.3	67.3	40.5	33.8	47.5	52.3	53.4
<i>Standard Calibration</i>									
CC	84.0	23.9	8.1	1.0	5.3	0.2	0.1	1.0	15.4
DC	80.6	23.5	25.2	1.0	9.0	0.2	0.1	2.1	17.7
GC	86.1	18.8	47.4	15.2	6.5	0.2	0.1	3.3	22.2
BC	86.9	33.2	49.9	1.0	0.4	0.2	2.8	3.4	22.2
<i>Normalized Calibration</i>									
NCC	78.0	38.2	65.1	92.5	67.7	33.8	59.5	73.1	63.5
NDC	83.8	44.2	63.6	92.3	68.8	32.2	64.4	75.5	65.6
NGC	83.6	44.6	61.2	90.8	67.7	22.4	56.9	69.2	62.1
NBC	83.8	49.5	64.9	92.0	69.8	32.8	61.1	73.4	65.9

Table 9: Comparison of different calibration methods with Llama 3.1 (8B). CC = contextual calibration (Zhao et al., 2021); DC = domain-context calibration (Fei et al., 2023); GC = generative calibration (Jiang et al., 2023b); BC = batch calibration (Zhou et al., 2024); NCC = normalized contextual calibration; NDC = normalized domain-context calibration; NGC = normalized generative calibration; NBC = normalized batch calibration.

context calibration (NDC), normalized generative calibration (NGC), and normalized batch calibration (NBC).

Given the computational cost of evaluating all methods across multiple models and datasets, we limit this comparison to the Llama 3.1 (8B) model on the text classification datasets.

The results are shown in Table 9. Overall, for standard calibration, DC, GC, and BC generally improve over CC on datasets where label length bias is limited (e.g., AG News, SST-5, Yahoo), but they still struggle when multi-token label bias is strong (e.g., DBpedia, 20 Newsgroups, TREC-50, Banking77, CLINC150), where NCC consistently performs best. For normalized calibration, augmenting these methods with length normalization (NDC/NGC/NBC) can yield further gains over NCC, but these improvements are smaller than the gaps between NCC and CC/DC/GC/BC, indicating that the main performance boost comes from combining some form of semantic calibration with length normalization.

E Reliability of the LLMs Confidences

As shown in Section 8, NCC improves the reliability of model confidence scores. Here, we extend that analysis by examining how often labels are predicted based on their length (left histograms) and how prediction confidences are distributed (right histograms). We also report the enrichment factor, which quantifies how much more frequently certain labels are predicted compared to the ground

truth distribution. Label lengths are normalized within each dataset, such that the shortest label corresponds to 0%, the longest to 100%, and intermediate labels are scaled proportionally. This analysis is conducted using the Llama 3.1 (8B) model across all classification datasets from the main text.

The results, summarized in Figure 11, further highlight the label length bias problem: the raw probabilities from the LLM strongly favor shorter labels, with an enrichment factor exceeding 3, and predicts them with high confidence (near certainty in over 30% of cases).

Length normalization (Norm Prob) reduces this bias by flattening probabilities across labels of different lengths, producing a confidence distribution skewed toward lower values (i.e., most predictions fall within the 10%–50% confidence range). However, as discussed in Section 3, it introduces a new issue: later tokens in longer labels become disproportionately likely, resulting in an over-prediction of long labels. Similarly, standard CC (without length normalization) overcompensates for the length bias, favoring longer labels instead, as analyzed in Appendix B.

In contrast, NCC produces label predictions that closely match the true distribution across label lengths, effectively mitigating both short- and long-label biases. Additionally, its prediction confidence is more evenly distributed and more closely resembles a well-calibrated model, reflected by the lowest ECE in the top plot of Figure 11.

Multi-token (Original)	Single-token (Remapping)
very positive	great
positive	good
neutral	neutral
negative	bad
very negative	terrible

Table 10: Mapping of the original SST-5 multi-token class labels to single token alternatives, as done in prior work.

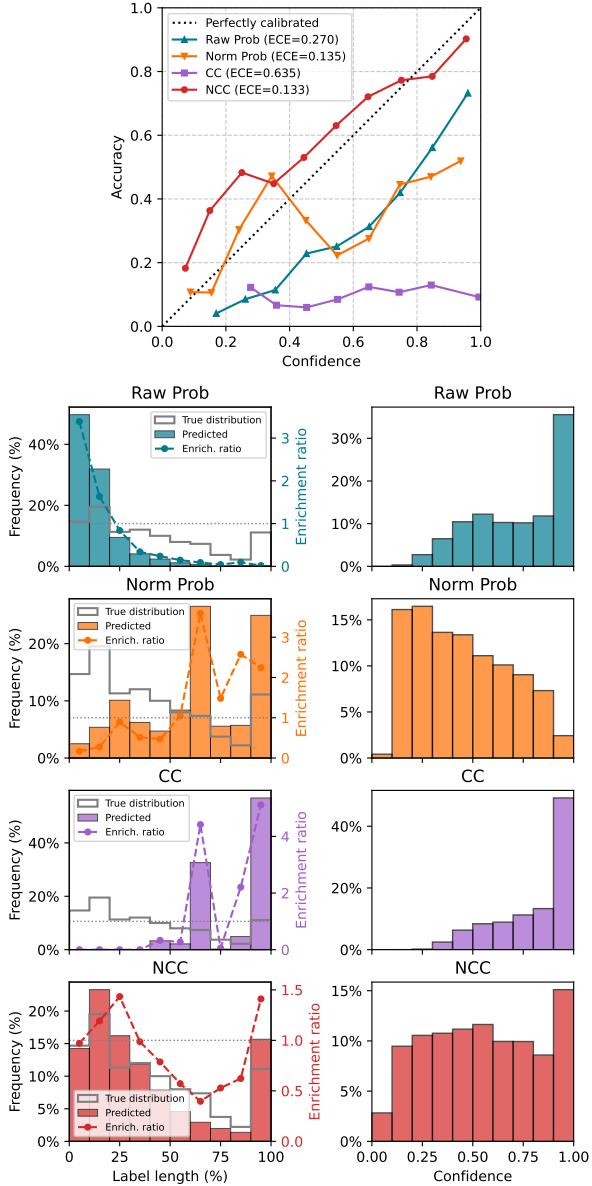


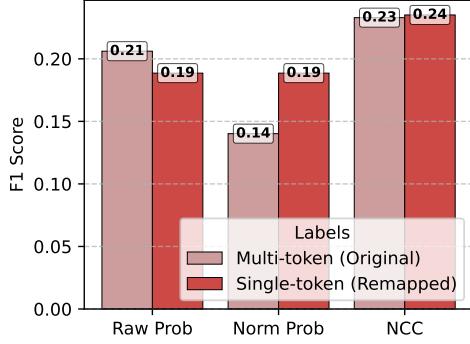
Figure 11: Reliability diagrams and frequency of label prediction by label length (left) and confidence (right).

F Multi-Token vs. Single-Token Labels

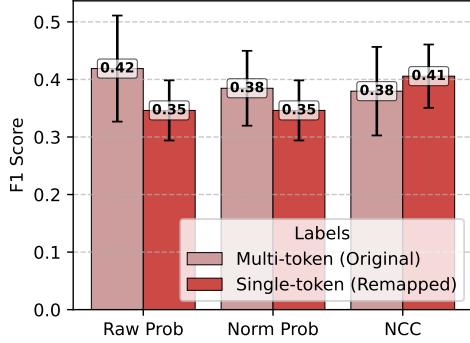
In the main text of the paper, we demonstrate improved label bias mitigation by applying NCC, a calibration technique at the full-label level to solve the identified label length bias when working with multi-token labels. Previous works do not consider this type of bias, as they focus on single-token labels by remapping original multi-token labels to single-token alternatives (Zhao et al., 2021; Fei et al., 2023; Han et al., 2023; Jiang et al., 2023b; Zhou et al., 2024). For instance, in sentiment analysis tasks, labels like “*very positive*” and “*very negative*” are often replaced by “*great*” and “*terrible*”, respectively. These label remappings are not always feasible, and it is ambiguous to determine when a label can be represented by a single token without the risk of losing semantic meaning. For example, in the Banking77 dataset (Casanueva et al., 2020) labels like “*balance not updated after cheque or cash deposit*” and “*top up failed*” contain nuances hardly distinguishable by means of an alternative single-token label.

However, one question that remains open is whether, when possible, considering single-token alternatives yields better results than the original multi-token labels. To address this, we conduct an additional analysis comparing the performance of the original multi-token labels against the widely adopted remappings of the SST-5 dataset (Socher et al., 2013). We consider the original labels and the single-token alternatives presented in Table 10. We run this analysis with the Llama 3.1 (8B) model in both zero- and few-shot settings. As in our main methodology, the few-shot setting is run with 5 different random seeds.

Figure 12 compares the standard raw probability approach, probability with length normalization, and NCC when using either the original multi-token labels or simplified single-token alternatives. In the standard Raw Prob setting, multi-token labels are slightly favored, though the difference re-



(a) Zero-shot ($k = 0$).



(b) Few-shot ($k = 5$).

Figure 12: Performance comparison of original multi-token labels against single-token alternatives.

mains within the error margins. A notable discrepancy emerges in the *zero-shot* setting with the Norm Prob method, where multi-token labels containing “*very X*”⁵ are over-predicted. This occurs because the token “*X*” following “*very*” receives higher conditional probability than when predicted alone, leading to inflated overall label scores – a problem also discussed in Section 3. In contrast, NCC performs consistently well across both label types, demonstrating its ability to mitigate this bias. These results indicate that retaining the original multi-token labels does not hinder performance and avoids potential semantic loss introduced by label simplification.

⁵Where “*X*” denotes either “*positive*” or “*negative*”.