Surprise Calibration for Better In-Context Learning

Zhihang Tan[1] **Jingrui Hou**[1*] **Ping Wang**[1,2*] **Qibiao Hu**[1] **Peng Zhu**[3]
[1]School of Information Management, Wuhan University
[2]Center for the Studies of Information Resources, Wuhan University
[3]School of Economics and Management, Nanjing University of Science and Technology
{zhihangtan, houjingrui, wangping, huqibiao}@whu.edu.cn, pzhu@njust.edu.cn

Abstract

In-context learning (ICL) has emerged as a powerful paradigm for task adaptation in large language models (LLMs), where models infer underlying task structures from a few demonstrations. However, ICL remains susceptible to biases that arise from prior knowledge and contextual demonstrations, which can degrade the performance of LLMs. Existing bias calibration methods typically apply fixed class priors across all inputs, limiting their efficacy in dynamic ICL settings where the context for each query differs. To address these limitations, we adopt implicit sequential Bayesian inference as a framework for interpreting ICL, identify "surprise" as an informative signal for class prior shift, and introduce a novel method—Surprise Calibration (SC). SC leverages the notion of surprise to capture the temporal dynamics of class priors, providing a more adaptive and computationally efficient solution for in-context learning. We empirically demonstrate the superiority of SC over existing bias calibration techniques across a range of benchmark natural language processing tasks. [1]

1 Introduction

In recent years, In-context learning (ICL) has emerged as a powerful paradigm to enable large language models (LLMs) to adapt to natural language processing tasks with minimal supervision (Radford et al., 2019; Liu et al., 2023; Dong et al., 2022; Zhou et al., 2024). Unlike traditional methods that rely on retraining, ICL enables models to adapt to new tasks by simply providing a set of demonstrations within the input context. While ICL has demonstrated significant success, it remains susceptible to biases, a phenomenon reflecting LLMs' tendency to make predictions that favor certain categories over others (Min et al., 2022; Holtzman et al., 2021; Levine et al., 2021; Si et al., 2023). These biases can degrade the performance of the LLMs, For example, if a class prior overemphasizes one category, the model may be more likely to predict that category, even when it's not the most relevant or accurate choice. This issue has spurred research into developing methods to calibrate biases within ICL (Zhao et al., 2021; Han et al., 2022; Zhou et al., 2023; Abbas et al., 2024). However, most existing methods perform a one-time calibration of class priors and apply them uniformly across all inputs. While such techniques outperform vanilla ICL, they fall short in dynamic ICL scenarios, where each input instance may be paired with distinct contextual demonstrations (Rubin et al., 2022; Zhang et al., 2022; Li and Qiu, 2023; Shu and Du, 2024; Luo et al., 2024). As a result, such techniques cannot adapt efficiently to the changing context.

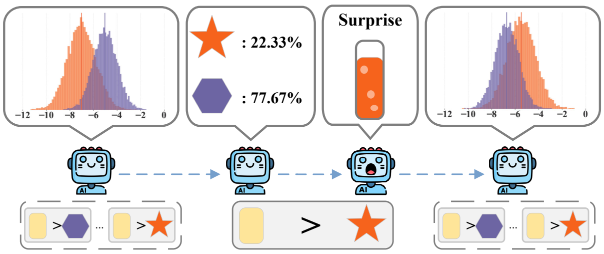


Figure 1: Illustration of surprise signal. The yellow rectangle denotes the example input in ICL, while the star and hexagon represent different class labels. The ">" symbol indicates the separator. The colored histograms represent the distribution of the model's predicted log probabilities for each class at that step. A mismatch between the model's prediction and the ground-truth label leads to surprise, which in turn causes a shift in the class prior.

In this work, we adopt implicit sequential Bayesian inference as a lens to interpret ICL, modeling it as a process in which LLMs continually update their implicit beliefs upon observing new demonstrations. Under this perspective, we propose the notion of *surprise* as a potential

[*]Corresponding authors.
[1]The code is available on GitHub: https://github.com/yan-muzhou/Surprise-Calibration.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 23057–23072
November 4-9, 2025 ©2025 Association for Computational Linguistics

signal for prior shift, which quantifies how "surprised" LLMs are when encountering new demonstrations(as shown in Figure 1). And our empirical analysis verifies that surprise functions as a robust and effective mechanism for class prior adjustment, even when other contextual biases are present.

Based on these insights, we propose a novel method: Surprise Calibration (SC). SC integrates surprise as a signal for detecting a shift in class prior. Rather than relying on fixed priors, SC dynamically updates class priors based on the evolving context, allowing the model to adapt its predictions in the context of dynamic ICL.

We evaluate the effectiveness of SC across eight datasets from six NLP tasks. Our experiments demonstrate that SC outperforms state-of-the-art ICL bias calibration baselines and shows superior robustness and stability.

The contributions of this paper are as follows:

- We reinterpret in-context learning through the lens of implicit sequential Bayesian inference, and identify surprise as a core signal that drives class prior adjustment.
- We introduce Surprise Calibration, a novel approach to dynamically calibrating ICL based on the cumulative effect of surprise signals on class priors.
- We empirically demonstrate the superior performance and computational efficiency of SC on a range of NLP tasks, showing its potential for scalable and robust ICL in real-world applications.

2 Background

2.1 In-Context Learning

ICL refers to the ability of a LLM to adapt to new tasks or instructions by leveraging a set of provided demonstrations (prompts) without requiring explicit retraining (Radford et al., 2019; Liu et al., 2023; Dong et al., 2022). Typically, given a query (a test instance for which the model needs to generate a prediction) and context (a set of demonstrations), ICL leverages the knowledge encoded in the pre-trained model parameters and the contextual information to generate an answer (Brown et al., 2020).

Formally, let $\mathcal{E}$ denote the space of all possible input queries, and let $\mathcal{Y}$ represent the space of all possible outputs (i.e., labels). For a given query $e \in \mathcal{E}$, the model produces a predicted label $\hat{y}$ by evaluating the conditional probability distribution over all candidate labels $y' \in \mathcal{Y}$:

$$\hat{y} = \arg\max_{y' \in \mathcal{Y}} p(y' \mid e, D), \tag{1}$$

where D represents the context. Early implementations of ICL relied on a fixed set of demonstrations (either hand-crafted or randomly selected) for all potential queries (Hendrycks et al., 2021; Wei et al., 2022; Lewkowycz et al., 2022). In such setups, the context D consists of K examples, i.e., $D = \{(e_j, y_j\}_{j=1}^{K}$, where each pair (e_j, y_j) corresponds to an example, with $e_j \in \mathcal{E}$ and $y_j \in \mathcal{Y}$.

However, the performance of ICL is sensitive to the selection of demonstrations (Liu et al., 2022; Qin et al., 2023). Recent advancements have introduced methods for dynamically selecting query-specific demonstrations, which are collectively referred to as *dynamic ICL* (Qin et al., 2023; Wang et al., 2024; Peng et al., 2024). Unlike static demonstrations, dynamic ICL tailors the context D for each individual query e. One typical approach involves selecting the most relevant K examples to the query, optimizing model performance based on factors such as semantic similarity or task-specific relevance (Liu et al., 2022; Luo et al., 2024).

2.2 Bias of In-Context Learning

Despite its empirical success in few-shot learning, ICL has been shown to be susceptible to systematic biases that can distort predictions. Zhao et al. (2021) highlighted that the biases observed in LLMs predominantly stem from two key sources: the contextual demonstrations provided and the inherent characteristics of the model itself. Contextual bias encompasses issues such as majority label bias, where models tend to favor the most frequent label in the demonstration set, and recency label bias, where models show a preference for labels that appear at the end of the demonstration sequence. On the other hand, model-intrinsic bias reflects the model's inherent propensity to predict label tokens that were frequent in its pretraining corpus. These biases can significantly undermine the robustness and reliability of ICL methods, leading to suboptimal performance in real-world applications.

To mitigate these biases, prior work (Zhao et al., 2021; Han et al., 2022; Fei et al., 2023; Zhou et al., 2023; Abbas et al., 2024) has suggested calibrating the predicted probabilities of LLMs by incorporating class prior probabilities, which can be estimated through repeated sampling. Formally, the prior probability for any label $y' \in \mathcal{Y}$ can then be expressed as:

$$p(y') = p(y' \mid D). \tag{2}$$

In practice, estimating this prior probability typically involves repeated sampling. For a fixed context D, the sampling process is repeated n times (e.g., with a fixed value $n = 3$, as used in Zhao et al. (2021)). However, explicitly accounting for contextual bias by performing separate prior estimation for each query can lead to prohibitively high computational costs. Conversely, disregarding contextual bias entirely may result in suboptimal performance, as it fails to capture critical variations in the prior distribution. This trade-off is particularly pronounced in *dynamic ICL* settings, where the context D varies for each query, making repeated sampling impractical for real-time or resource-constrained applications.

This work seeks to address the aforementioned challenge by proposing an efficient method for estimating class priors in dynamic ICL scenarios. Unlike static ICL, accurately estimating the prior in dynamic settings requires a comprehensive understanding of the process by which ICL models utilize demonstrations for inference. To this end, we first establish a theoretical framework to model this process and subsequently introduce a novel approach that enhances the accuracy and robustness of dynamic ICL.

3 Surprise: a Signal for Class Prior Shift

3.1 Reinterpretation ICL via Implicit Sequential Bayesian Inference

We first highlight two essential characteristics of LLM that motivate our interpretation of in-context learning as implicit sequential Bayesian inference. **(1) Latent concept.** In the context of ICL, prior works have proposed that transformers implicitly infer a latent concept z underlying the demonstration data D, which is subsequently leveraged to guide predictions (Xie et al., 2021; Tefnik and Kadlcik, 2023; Hendel et al., 2023; Han et al., 2024; Todd et al., 2024). Here, $z \in \mathcal{Z}$, where $\mathcal{Z}$ denotes the space of all possible concepts that explain the demonstrations, such as label assignment rules, task definitions, or underlying generative functions, depending on the task. Given this latent representation, the probability of predicting a label for a new input is modeled by marginalizing over all possible concepts:

$$p(y \mid e, D) = \int_{\mathcal{Z}} p(y \mid e, z)\, p(z \mid D)\, dz. \tag{3}$$

(2) Autoregressive language models and surprise signals. Large language models (LLMs) generate output autoregressively, one token at a time (Kossen et al., 2024). Given an input sequence $(X_1, \ldots, X_M)$, each token X_i is predicted based on its preceding context $(X_1, \ldots, X_{i-1})$. In ICL, prompts often structure each demonstration as: input e, followed by a label delimiter (e.g., '>' or ':'), then the corresponding label y. As the model processes these demonstrations sequentially, the hidden state at the delimiter captures its current belief about the label for e_j, conditioned on all previous $j-1$ demonstrations D_{j-1}—formally denoted as $p(y' \mid e_j, D_{j-1})$.

Crucially, this architecture allows us to measure the model's predictive uncertainty before observing each label. Specifically, the conditional probability $p(y_j \mid e_j, D_{j-1})$, computed just before the true label y_j is revealed, reflects how "surprised" the model is by the label. We define *surprise* as the negative log probability of the observed label:

$$\text{Surprise}(y_j \mid e_j, D_{j-1}) = -\log p(y_j \mid e_j, D_{j-1}). \tag{4}$$

Higher values indicate greater surprise, signaling a mismatch between the model's expectations and the observed label.

Surprise-driven bayesian updating. Building upon this intuition, we formalize the implicit Bayesian updating procedure based on observed demonstrations. Formally, let the model's belief over z prior to observing a new example be $p(z \mid D_{j-1})$. Incorporating (e_j, y_j) leads to a Bayesian update:

$$p(z \mid D_{j-1} \cup \{(e_j, y_j)\}) = \frac{p(e_j, y_j \mid z)\, p(z \mid D_{j-1})}{\int_{\mathcal{Z}} p(e_j, y_j \mid z)\, p(z \mid D_{j-1})\, dz}. \tag{5}$$

Given this update, the revised class prior for an arbitrary label y' becomes:

$$\begin{aligned} p(y' \mid D_{j-1} \cup \{(e_j, y_j)\}) =& \mathbb{E}_{z \sim p(z \mid D_{j-1})}[p(y' \mid z)] \\ &+ \frac{\mathrm{Cov}_{z \sim p(z \mid D_{j-1})}(p(y' \mid z),\, p(e_j, y_j \mid z))}{\mathbb{E}_{z \sim p(z \mid D_{j-1})}[p(e_j, y_j \mid z)]} \end{aligned} \tag{6}$$

Here, the denominator represents the expected joint likelihood of (e_j, y_j) under the current belief over

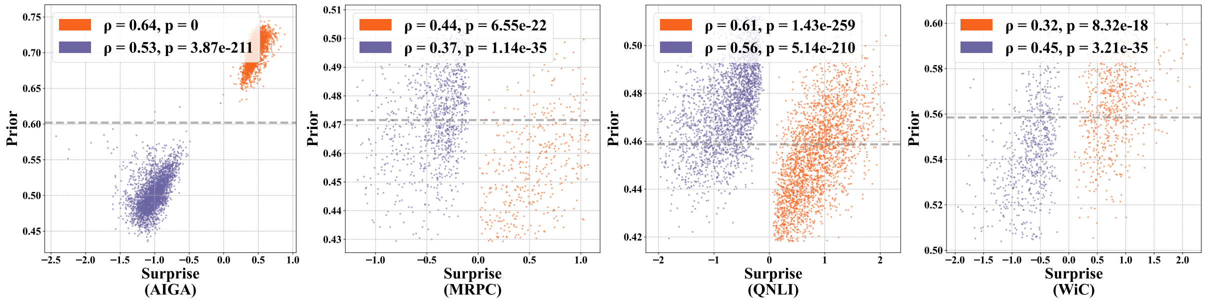


Figure 2: Spearman correlation between surprise ($-\log p(y|e,D)$) and the prior probability of the positive class across four binary classification datasets (AIGA, MRPC, QNLI, WiC). Each scatter plot shows the relationship of surprise and positive class prior for positive (orange) and negative (purple) demonstration insertions, with corresponding Spearman ρ and p-values. The gray dashed line marks the estimated prior before insertion. All priors are estimated by repeated sampling using the BC method (described in Section 5.1).

z. Comparing this to the predictive probability:

$$\begin{aligned} p(y_j \mid e_j, D_{j-1}) &= \mathbb{E}_{z\sim p(z|D_{j-1})}\left[p(y_j \mid e_j, z)\right] \\ &= \mathbb{E}_{z\sim p(z|D_{j-1})}\left[\frac{p(e_j, y_j \mid z)}{p(e_j \mid z)}\right]. \end{aligned} \tag{7}$$

We note that a low value of $p(y_j \mid e_j, D_{j-1})$ may suggest lower $\mathbb{E}_{z\sim p(z|D_{j-1})}[p(e_j, y_j \mid z)]$, though this relationship is not formally guaranteed due to the normalization by $p(e_j \mid D_{j-1})$. Nevertheless, we hypothesize that surprise—captured via $p(y_j \mid e_j, D_{j-1})$—can serve as a practical and meaningful signal for class prior adjustment. In particular, higher surprise(smaller value of $p(y_j \mid e_j, D_{j-1})$) is expected to amplify the influence of the covariance term, leading to larger shift in the predicted distribution over labels.

In summary, we highlight two possible mechanisms driving prior updates:

- **Covariance-driven adjustment:** Positive covariance between $p(y' \mid z)$ and $p(e_j, y_j \mid z)$ (i.e. $y' = y_j$) increases the posterior probability of y', while negative covariance(i.e. $y' \neq y_j$) decreases it (See Eq (6)).

- **Surprise Amplification via Joint Likelihood:** Higher surprise (corresponding to a lower average joint likelihood) enhances the sensitivity of the update to variations in z, thereby amplifying belief shifts (See Eq (6) and Eq (7)).

3.2 Empirical Evidence

To empirically validate that **surprise serves as a practical signal of prior shift**, we conducted controlled experiments on four binary classification datasets. For each dataset, we specifically focused on observing changes in the prior probability of a designated positive class upon inserting new examples.

By introducing a new example (e, y) into a fixed set of demonstrations, we recorded the following information:

- the prior probability of the positive class both before and after the insertion;

- the surprise value associated with the example, quantified as $-\log p(y \mid e, z)$.

Following the covariance-driven Bayesian updating framework, surprise values were assigned negative signs for negative-class examples and positive signs for positive-class examples, reflecting their respective suppressive and reinforcing effects on the positive class prior. By repeating this process with different examples added to the fixed context, we collected multiple data points, each consisting of the surprise value and its corresponding positive class prior.

Figure 2 presents the relationship between surprise and prior shift across datasets. Crucially, across all experimental setups, we consistently observed statistically significant positive Spearman correlations between surprise and the shift in the positive-class prior. Moreover, we observe that as the absolute value of surprise for positive-class examples increases, the class prior shifts toward the positive class. Conversely, as the absolute value of surprise for negative-class examples increases, the class prior shifts toward the negative class (manifested in the figure as a decrease in the positive-class prior). This result robustly supports our theoretical assertion that surprise systematically covaries with prior adjustments, effectively predicting

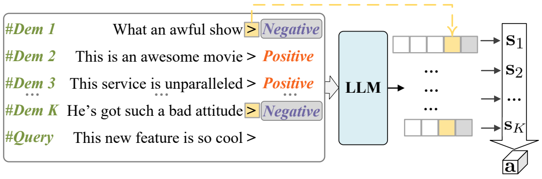


Figure 3: Illustration of the proposed Surprise Calibration for In-context learning.

the direction and magnitude of these shifts.

Importantly, this correlation was most pronounced within groups conditioned on the newly inserted example's label. Aggregate correlations occasionally showed instability, which we attribute to an additional influence—*anti-recency bias*. This bias tends to shift overall predictions away from recently observed labels, creating an offset in mean prior probabilities between groups. We further investigate the nature of this bias in Figure 9 in the Appendix.

Despite the presence of anti-recency bias, our findings emphasize a critical distinction: while biases may alter the overall mean prior levels, the structural relationship between surprise and prior adjustment remains robust and monotonic. Surprise reliably signals local updates to the model's beliefs about class prior.

4 Surprise Calibration

Building on the insights from Section 3, we propose **SC**, a novel calibration method that explicitly models the temporal propagation of surprise signals during ICL as illustrated in Figure 3.

To operationalize surprise-driven calibration, we begin by constructing a **surprise vector** for each in-context demonstration. Given a context example (e_j, y_j) and the model's current belief (conditioned on prior context D_{j-1}), we define the surprise vector $\mathbf{s} \in \mathbb{R}^C$ (C = number of classes), where the c-th dimension is

$$s_c = \bigl(1 - 2\,\delta_{c,y}\bigr)\,\log p\bigl(y = c \mid e_j,\, D_{j-1}\bigr). \tag{8}$$

where

$$\delta_{c,y} = \begin{cases} 1, & c = y, \\ 0, & c \neq y. \end{cases}$$

Here, the sign of s_c indicates whether class c matches the true label ($c = y$: negative, $c \neq y$: positive), and the magnitude reflects how surprising the label is to the model. The sign encodes the expected direction of prior adjustment for that class.

By collecting surprise vectors for each context example, we form a **surprise sequence**, i.e., $\mathcal{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_K]$, which explicitly represents both the amount and direction in which each demonstration should modulate the model's evolving class prior.

To capture dependencies across this sequence and aggregate the adjustment signals, we use a time series prediction model (e.g. GRU) to process the surprise sequence. The final hidden state of the GRU is decoded into a **prior adjustment vector** $\mathbf{a} \in \mathbb{R}^C$, which models the cumulative shift in class priors based on the observed context.

For a query input e, we compute the model's original prediction probability over label y': $p_{\text{orig}}(y' \mid e, z)$. Our calibrated probability is then obtained by adjusting the original log-probability with the corresponding prior adjustment:

$$-\log p_{\text{calib}}(y' \mid e, z) = -\log p_{\text{orig}}(y' \mid e, z) - a_{y'} \tag{9}$$

where $a_{y'}$ is the adjustment for the class y' inferred from the surprise sequence. This effectively shifts the model's confidence in the predicted class according to the accumulated surprise from previous in-context demonstrations.

The model is trained end-to-end by minimizing the cross-entropy loss between the calibrated probabilities and the true label of the query input, jointly optimizing the parameters of the time series prediction model and decoding layers.

5 Experiments

5.1 Experimental Setup

Datasets We evaluate the effectiveness of our SC method on 8 datasets within 6 natural language tasks. Specifically, we consider sentiment classification: SST-2 (Socher et al., 2013); natural language inference and entailment: RTE, QNLI, MNLI (Williams et al., 2018); paraphrasing: MRPC (Dolan and Brockett, 2005); word disambiguation: WiC (Pilehvar and Camacho-Collados, 2019); spam detection: YouTube Spam(YouTube) (Alberto et al., 2015); AIGC detection: AIGA (Theocharopoulos et al., 2023).

Models We conducted experiments on two pretrained models with different sizes from the Qwen2.5 series (Yang et al., 2024), Qwen2.5-3B, and Qwen2.5-7B. The Qwen2.5 series has shown highly competitive performance, matching or surpassing other models with similar parameter counts.

Baselines We compare our method with five advanced calibration methods: **ICL**: Vanilla in-context learning performance; **BC** (Batch Calibration) (Zhou et al., 2023): Estimates class priors directly from a batch of input data; **LinC** (Linear Probe Calibration) (Abbas et al., 2024): Estimates class priors using a labeled training dataset; **CC+** (Contextual Calibration) (Zhao et al., 2021): Estimates class priors per query via content-free examples; **BC+** and **LinC+**: Query-specific extensions of BC and LinC, respectively. The "+" symbol indicates that class priors are estimated individually for each query, which introduces additional inference overhead as shown in Table 1.

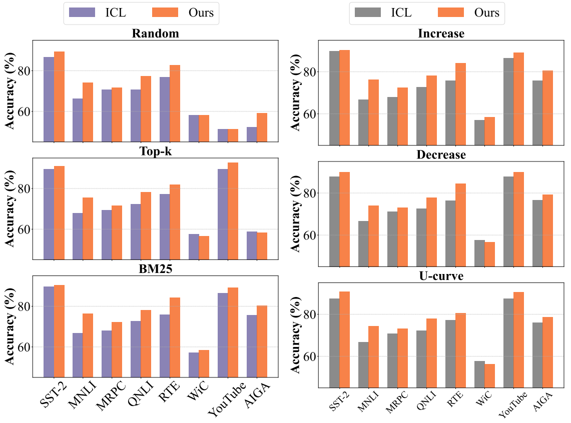
Method	Inference Count
BC	T
LinC	$M + T$
CC+	$3 \times T$
BC+	$n \times T$
LinC+	$n \times T$
Ours	$M + T$

Table 1: Comparison of Inference Count for Different Methods. T represents the number of samples to be predicted, n denotes the additional sample size used for estimating class priors in BC+ and LinC+, and M is the sample size used for training the SC and LinC model.

Demonstrations Selecting and Ordering Strategies We explore three demonstration selection strategies: **Random**, which randomly selects contextual examples; **BM25**, which uses word-overlap similarity to choose high-similarity demonstrations; and **Top-k**, which selects nearest neighbors based on cosine similarity of embeddings generated by the GTE model(Li et al., 2023). For BM25 and Top-k, we apply three ordering strategies: **Increase**, placing higher-value examples at the end; **Decrease**, placing them at the beginning; and **U-curve**, positioning higher-value examples at both ends and lower-value ones in the middle.

5.2 Main Experimental

Table 2 presents three main findings. Firstly, the proposed method, SC, consistently exhibits high performance across various model sizes and datasets. Specifically, SC outperforms ICL by +4.59% on Qwen2.5-3B and +3.54% on Qwen2.5-7B on average and outperforms or matches the best-performing baseline in most cases. Secondly, in contrast to other dynamic prior estimation techniques like CC+ and BC+, SC consistently outperforms ICL without any performance degradation in all settings. Lastly, SC achieves these results while maintaining computational efficiency, as it does not require additional multiplicative inference iterations to estimate class priors for each input query. Instead, it only requires a limited number of training samples to calibrate the priors, as Table 1 shows.



(a) 3 demonstration selection. (b) 3 demonstration ordering.

Figure 4: Accuracy comparison between SC and ICL across 3 demonstration selection and demonstration ordering strategies with other settings kept consistent as shown in Table 2.

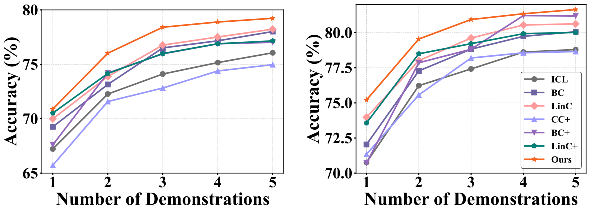


Figure 5: Average performance across 8 datasets varies with the number of demonstrations, with other settings kept consistent as shown in Table 2. The left panel displays results for Qwen2.5-3B, while the right panel shows results for Qwen2.5-7B.

5.3 Stability Analysis

Previous studies (Xie et al., 2021; Rubin et al., 2022; Liu et al., 2022; Lu et al., 2022; Wu et al., 2023; Guo et al., 2024; Zhang et al., 2024) have highlighted the sensitivity of ICL to the organization of demonstrations. In this section, we demonstrate that SC can enhance performance stability across a variety of scenarios.

Varying Numbers of Demonstrations As shown in Figure 5, SC again outperforms all baseline methods. Notably, we observe that the perfor-

DataSet	LM	Method						
		ICL	BC	LinC	CC+	BC+	LinC+	Ours
SST-2	Qwen2.5-3B	89.68	89.79 (+0.11)	89.90 (+0.22)	88.58 (-1.10)	89.84 (+0.16)	**90.33** (+0.65)	89.68(+0.00)
	Qwen2.5-7B	94.45	94.51 (+0.06)	94.51 (+0.06)	**95.00** (+0.55)	94.50 (+0.05)	94.67 (+0.22)	**95.00**(+0.55)
MNLI	Qwen2.5-3B	66.77	72.38 (+5.61)	73.44 (+6.67)	70.18 (+3.41)	**79.07** (+12.30)	67.52 (+0.75)	76.33 (+9.56)
	Qwen2.5-7B	67.01	75.03 (+8.02)	77.27 (+10.26)	79.78 (+12.77)	75.02 (+8.01)	67.60 (+0.59)	**81.15** (+14.14)
MRPC	Qwen2.5-3B	68.00	67.42 (-0.58)	67.77 (-0.23)	71.71 (+3.71)	71.71 (+3.71)	70.66 (+2.66)	**72.28** (+4.28)
	Qwen2.5-7B	68.17	67.82 (-0.35)	72.35 (+4.18)	73.33 (+5.16)	73.97 (+5.80)	72.86 (+4.69)	**73.85** (+5.68)
QNLI	Qwen2.5-3B	72.67	76.57 (+3.90)	76.51 (+3.84)	69.47 (-3.20)	**79.53** (+6.86)	79.26 (+6.59)	78.01 (+5.34)
	Qwen2.5-7B	80.26	80.34 (+0.08)	80.19 (-0.07)	74.48 (-5.78)	80.83 (+0.57)	**81.01** (+0.75)	80.48 (+0.14)
RTE	Qwen2.5-3B	75.81	82.67 (+6.86)	83.03 (+7.22)	72.93 (-2.88)	75.45 (-0.36)	75.81 (+0.00)	**84.47** (+8.66)
	Qwen2.5-7B	83.39	**83.75** (+0.36)	**83.75** (+0.36)	79.06 (-4.33)	81.22 (-2.17)	83.03 (-0.36)	**83.75** (+0.36)
WiC	Qwen2.5-3B	57.07	**59.57** (+2.50)	58.93 (+1.86)	54.93 (-2.14)	**59.57** (+2.50)	58.71 (+1.64)	58.29 (+1.22)
	Qwen2.5-7B	62.86	63.50 (+0.64)	62.93 (-0.07)	60.57 (-2.29)	63.21 (+0.35)	**63.92** (+1.06)	62.93 (+0.07)
YouTube	Qwen2.5-3B	86.48	86.73 (+0.25)	87.24 (+0.76)	78.83 (-7.65)	77.04 (-9.44)	88.52 (+2.04)	**90.56** (+4.08)
	Qwen2.5-7B	87.50	88.52 (+1.02)	88.52 (+1.02)	84.18 (-3.32)	80.87 (-6.63)	88.27 (+0.77)	**90.56** (+3.06)
AIGA	Qwen2.5-3B	76.37	76.97 (+0.60)	77.33 (+0.96)	75.91 (-0.46)	75.47 (-0.90)	77.12 (+0.75)	**79.99** (+3.62)
	Qwen2.5-7B	75.71	77.20 (+1.49)	77.40 (+1.69)	79.16 (+3.45)	80.93 (+5.22)	82.43 (+6.72)	**79.97** (+4.26)
Avg.	Qwen2.5-3B	74.11	76.51 (+2.40)	76.77 (+2.66)	72.82 (-1.29)	75.96 (+1.85)	75.99 (+1.88)	**78.70** (+4.59)
	Qwen2.5-7B	77.42	78.83 (+1.41)	79.62 (+2.20)	78.20 (+0.78)	78.82 (+1.40)	79.22 (+1.80)	**80.96** (+3.54)

Table 2: Accuracy(%) comparison of different calibration methods on various datasets using BM25 selecting strategy, increase ordering strategy and 3-shot Qwen2.5 models (3B and 7B). We train a GRU model as the backbone of SC framework and report results using fixed random seeds and hyper-parameters. The best performance for each dataset and model size is highlighted in bold. The percentage changes relative to ICL are shown in parentheses. Green indicates improvement, and red indicates a reduction.

mance of SC under the 2-shot setting can match or even exceed that of other calibration methods in the 3-shot setting or Vanilla ICL in the 5-shot setting. Considering that the inference cost of ICL is proportional to the context length, SC can significantly reduce computational resource consumption while maintaining comparable performance, making it particularly advantageous for real-time inference scenarios with growing demands.

Varying Demonstration Selecting and Ordering Strategies To comprehensively validate that SC can robustly enhance the performance of ICL, we examine both demonstration selection and ordering strategies. Specifically, we evaluate three distinct demonstration selection strategies and three ordering strategies, as outlined in Section 5.1. As illustrated in Figure 4, SC demonstrates superior performance compared to Vanilla ICL across most scenarios for both selection and ordering. Our findings indicate that selecting demonstrations more similar to the test samples generally yields better results than exclusively choosing dissimilar ones, consistent with insights reported by Liu et al. (2022). Furthermore, the evaluation of ordering strategies confirms the effectiveness and reliability of SC in enhancing model performance, as it outperforms Vanilla ICL in nearly all cases.

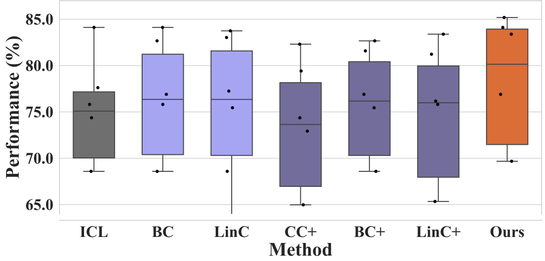


Figure 6: Performance comparison of calibration methods on RTE dataset, with other settings kept consistent as shown in Table 2.

Varying Verbalizers We investigate the robustness of SC to variations in verbalizer designs and find that different verbalizers influence model performance as shown in Figure 6. We observed that irrespective of the verbalizer configuration, our method consistently achieves the best performance, with a median accuracy exceeding 80%.

Impact of Training Set Size Figure 7 examines the impact of additional training samples on the performance of the RTE dataset when using Qwen2.5-3B under 1-shot, 3-shot, and 5-shot settings. The findings reveal that the model's performance remains robust across a broad range of sample sizes,

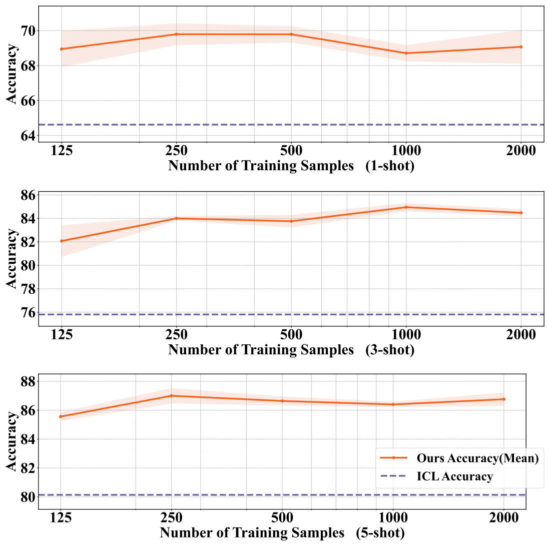


Figure 7: Performance across varying numbers of training samples under 1-shot, 3-shot, and 5-shot settings, averaged over three runs with fixed random seedse, with other settings kept consistent as shown in Table 2.

	SST-2	MNLI	MRPC	QNLI
w/	89.68 ± 0.37	76.33 ± 0.44	72.30 ± 0.07	78.18 ± 0.13
w/o	88.91 ± 0.44	77.02 ± 0.34	71.88 ± 0.20	77.32 ± 0.35
	RTE	WiC	YouTube	AIGA
w/	84.44 ± 0.16	58.05 ± 0.34	90.56 ± 0.12	79.98 ± 0.04
w/o	83.75 ± 0.18	56.64 ± 0.28	89.71 ± 0.24	78.00 ± 0.18

Table 3: Ablation Study. Comparison of accuracy(%) with (w/) and without (w/o) surprise magnitude, with other settings kept consistent as shown in Table 2. Results are reported as the mean ± standard deviation over three runs with three fixed random seeds.

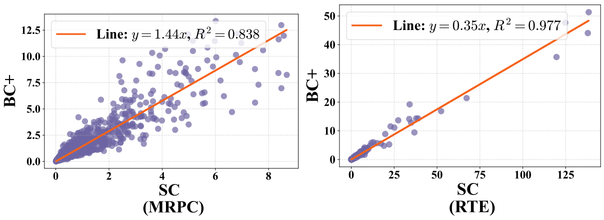


Figure 8: Scatter plot comparing the calibrated probability ratios of SC and BC on the MRPC (left) and RTE (right) datasets. Each point represents a single evaluation instance, with other settings kept consistent as shown in Table 2.

consistently surpassing the ICL baseline. Furthermore, as the number of shots increases, the calibration model demonstrates greater stability. We hypothesize that a higher number of shots provides more diverse signals, which enhances the overall signal-to-noise ratio, thereby improving stability.

5.4 Ablation Studies and Effectiveness Analysis

Impact of Surprise Amplification To investigate the impact of surprise magnitude on the calibration performance of SC, we conducted ablation experiments by rescaling each dimension of the surprise vector to have an absolute value of 1, while keeping its original sign. The experimental results in Table 3 indicate that incorporating the magnitude of surprise yields clear performance improvements for SC in most scenarios, suggesting that surprise magnitude plays a crucial role in enabling SC to accurately estimate class priors.

Effectiveness Analysis To verify whether the trained SC model truly learns class priors through surprise-driven signals, we compare the calibrated probability ratio[2] which directly reflects the model's relative belief in the two classes between our SC method and the Batch Calibration (BC) method.

As shown in Figure 8, there is a strong positive linear relationship between the calibrated probability ratios of SC and BC across both datasets (MRPC: $R^2 = 0.838$, RTE: $R^2 = 0.977$). This indicates that the SC model successfully captures the pattern of prior adjustment from the surprise sequence, and learns a linear decision boundary closely aligned with BC. The linear regression slope deviates from 1, which can be attributed to the fact that both methods rely on a particular subset of data to estimate priors; differences in sample distributions can introduce a global bias, resulting in a shifted slope.

6 Conclusion

This work adopt implicit sequential Bayesian inference as a framework for interpreting ICL, where each new demonstration is treated as an update to the prior knowledge. This interpretation allows for a more formal and principled calibration of class priors. Based on this perspective, we introduced Surprise Calibration (SC), a novel approach to enhancing the performance and stability of In-context learning (ICL) across diverse natural language tasks.

Unlike existing methods, SC does not require additional inference iterations for each input query. Instead, it uses a small number of training sam-

[2] Given the calibrated probabilities $p_{\text{calib}}(y = 1|e)$ and $p_{\text{calib}}(y = 0|e)$, the ratio is defined as $\frac{p_{\text{calib}}(y=1|e)}{p_{\text{calib}}(y=0|e)}$.

ples to calibrate priors, offering substantial reductions in computational cost without sacrificing performance. We evaluated our method across eight datasets in six natural language tasks. Our experiments demonstrate that SC outperforms state-of-the-art ICL bias calibration baselines. Additionally, we demonstrate that ICL effectively captures the temporal dynamics of class priors, providing more effective and robust solutions for a wide range of ICL applications.

Limitations

This work has several limitations. First, although our calibration performance is more stable compared to other methods, there may still be a slight decrease in performance compared to vanilla ICL in very rare cases. We speculate that this may be because implicit sequential Bayesian inference does not fully capture the behavior of ICL, rendering the surprise signal less effective in these exceptional instances. Besides, our method requires a certain amount of labeled data, making it difficult to apply in environments with extremely limited resources. Lastly, our calibration model structure is relatively simple and can be easily affected by data noise caused by factors such as model inference accuracy.

Acknowledgment

This work is supported by the National Science and Technology Major Project (No. 2023ZD0121502), the National Social Science Foundation of China [No. 24&ZD186], the National Natural Science Foundation of China [Nos. 72374161, 72474160, 72474103 and 72174087], the China Postdoctoral Science Foundation (No. 2025M773205), and the Postdoctoral Fellowship Program of CPSF (No. GZC20252319).

References

Momin Abbas, Yi Zhou, Parikshit Ram, Nathalie Baracaldo, Horst Samulowitz, Theodoros Salonidis, and Tianyi Chen. 2024. Enhancing in-context learning via linear probe calibration. In *International Conference on Artificial Intelligence and Statistics*, pages 307–315. PMLR.

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. 2024. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966.

Túlio C Alberto, Johannes V Lochter, and Tiago A Almeida. 2015. Tubespam: Comment spam filtering on youtube. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 138–143. IEEE.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

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 (ACL 2023): Long Papers, Vol 1*, pages 14014–14031.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904.

Seungwook Han, Jinyeop Song, Jeff Gore, and Pulkit Agrawal. 2024. Emergence of abstractions: Concept encoding and decoding mechanism for in-context learning in transformers. *arXiv preprint arXiv:2412.12276*.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2022. Prototypical calibration for few-shot learning of language models. *arXiv preprint arXiv:2205.10183*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

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.

Jannik Kossen, Yarin Gal, and Tom Rainforth. 2024. In-context learning learns label relationships but is not conventional learning. In *The Twelfth International Conference on Learning Representations*.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2021. The inductive bias of in-context learning: Rethinking pretraining example design. *arXiv preprint arXiv:2110.04541*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Xiaonan Li and Xipeng Qiu. 2023. Finding supporting examples for in-context learning. *CoRR*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

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.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

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.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024. In-context learning with retrieved demonstrations for language models: A survey. *Preprint*, arXiv:2401.11624.

Sewon Min, Xinxi 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.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2024. Revisiting demonstration selection strategies in in-context learning. *arXiv preprint arXiv:2401.12087*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273.

Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. 2023. In-context learning with iterative demonstration selection. *arXiv preprint arXiv:2310.09881*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671.

Dong Shu and Mengnan Du. 2024. Comparative analysis of demonstration selection algorithms for llm in-context learning. *arXiv preprint arXiv:2410.23099*.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Michal Tefnik and Marek Kadlcik. 2023. Can in-context learners learn a reasoning concept from demonstrations? In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Panagiotis C. Theocharopoulos, Panagiotis Anagnostou, Anastasia Tsoukala, Spiros V. Georgakopoulos, Sotiris K. Tasoulis, and Vassilis P. Plagianakos.

2023. Detection of fake generated scientific abstracts. In *2023 IEEE Ninth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 33–39.

Eric Todd, Millicent Li, Arnab Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *International Conference on Learning Representations*. ICLR.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 1112–1122. Association for Computational Linguistics (ACL).

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

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, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148.

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2024. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems*, 36.

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.

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 preprint arXiv:2309.17249*.

Yuxiang Zhou, Jiazheng Li, Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. 2024. The mystery of in-context learning: A comprehensive survey on interpretation and analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14365–14378.

A Appendix

A.1 Dataset Statistics

Dataset	Sentences	Classes	|Train|	|Test|	Verbizer
SST-2	1	2	5000	1821	negative/positive
MNLI	2	3	15000	9815	no/maybe/yes
MRPC	2	2	4075	1724	no/yes
QNLI	2	2	5000	5463	no/yes
RTE	2	2	2490	277	no/yes
WiC	3	2	5428	1400	false/true
YouTube	1	2	1564	392	truthful/deceptive
AIGA	1	2	4320	5732	0/1

Table 4: Details of the dataset used for evaluation,"Sentences" denotes the number of segments in the input, while "Classes" refers to the number of categories in the label space, "|Train|" and "|Test|" denotes the number of test samples. When the labels of the test set are not publicly available, we use the validation set as the test set. For all datasets, we use a verbalizer that is semantically aligned with its label space.

A.2 Implementation Details

- **BC** (Batch Calibration) (Zhou et al., 2023): We reproduce the BC baseline by using the batch of all testing samples to estimate the class prior.

- **LinC** (Linear Probe Calibration) (Abbas et al., 2024): We follow the original implementation of LinC and using the same training sample size as SC method.

- **CC+** (Contextual Calibration) (Zhao et al., 2021): We adhere to the original implementation of CC and compute the mean of the log-probabilities over three content-free tokens—'N/A', '', and '[MASK]'—as the test sample within the predefined template.

- **BC+** We extend the BC baseline by using a fixed batch of labeled samples to estimate the class prior for each new input, selecting five samples for each class and combining them into a single batch.

- **LinC+** We extend the LinC baseline by using a fixed number of labeled samples to estimate the class prior for each new input, selecting five samples for each class and combining them into a training set.

- **SC** In our preliminary experiments, we evaluated several common backbone architectures, including GRU, LSTM, vanilla RNN, and Transformer Encoder. The results are summarized in the following table(Qwen2.5-3B;RTE;3-shot): As shown above, all backbones achieved very similar accuracy, indicating that the choice of backbone had minimal influence on the final calibration performance. We selected GRU due to its simpler structure and ease of training. The BSC model was trained for 200 epochs using the Adam optimizer with a learning rate of 1e-4. Hyperparameters were selected based on early-stage experiments on the MRPC dataset, using 20% of its training data as a validation set. These selected hyperparameters were then applied unchanged to all other datasets. Given the relatively simple nature of the model and datasets, we found this level of tuning to be sufficient.

Backbone	**Accuracy**
GRU	84.47
LSTM	83.95
RNN	84.47
Transformer	84.47

Table 5: Accuracy(%) comparison of different backbones.

- **Model Predicted Probabilities**: In the general practice of ICL, due to the influence of the in-context demonstrations, when the model predicts the label for a query, the sum of probabilities for all tokens in the label space tends to be very close to 1, while the probabilities of other tokens in the vocabulary are close to 0. Therefore, to understand the model's final prediction preference, we can focus solely on the limited tokens within the label space. Consequently, we only decode the hidden states of each delimiter into the tokens of the label space. (Only keep the logits of the label tokens.) So far, we have assumed that each label string is encoded as a single token. However, our approach can also be applied if some or all labels are encoded as multiple tokens. In essence, we continue to measure only the probability the model assigns to the first token of each label, making the (fairly harmless) assumption that the first (or only) token that each label is encoded to is unique among labels. We believe this is justified, as, given the first token for a label, the model should near-deterministically predict the remaining tokens, i.e. all the predictive information is contained in the first token the model predicts for a label. For example, for the YouTube dataset, the label 'truthful' is encoded by the Qwen tokenizer as two tokens, [truth, ful]. We only use the probability assigned to [truth] to assign probabilities to 'truthful', and ignore any predictions for [ful].

A.3 Additional Experiments

Figure 9 illustrates that the effect of anti-recency bias diminishes as the number of demonstrations. Table 6 illustrates the experimental results on more shot. Figure 5 illustrates the experimental results, providing evidence for the validity of the theorem introduced in Section 3.1. Table 7 summarizes the experimental outcomes under 1-shot, 2-shot, 4-shot, and 5-shot settings. Table 8 shows the additional results using LLama3-8B across all datasets. Figures 10 and 11 present a comparative analysis between the SC approach and the baseline method.

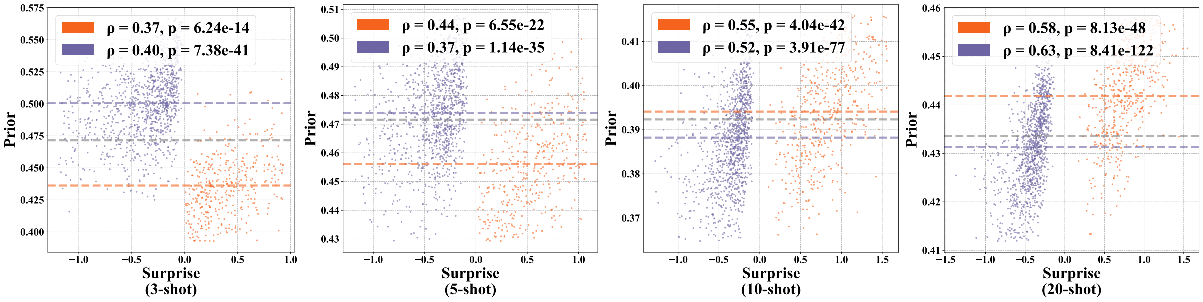


Figure 9: Spearman correlation between surprise ($-\log p(y|e, D)$) and the prior probability of the positive class across various in-context settings (3-, 5-, 10-, 20-shot) with MRPC dataset. Each scatter plot shows the relationship for positive (orange) and negative (purple) demonstration insertions, with corresponding Spearman ρ and p-values. The gray dashed line marks the estimated prior before insertion, while orange and purple dashed lines indicate the average prior after inserting positive and negative demonstration, respectively. All priors are estimated by repeated sampling using the BC method (described in Section 5.1). Results show the effect of anti-recency bias diminishes with increasing numbers of in-context demonstrations, indicating it is primarily small-sample phenomenon.

	10-shot Results							15-shot Results						
Dataset	**ICL**	**BC**	**LinC**	**CC+**	**BC+**	**LinC+**	**Ours**	**ICL**	**BC**	**LinC**	**CC+**	**BC+**	**LinC+**	**Ours**
MRPC	73.62	72.11	72.93	71.01	74.02	73.97	**73.86**	74.38	72.05	73.91	70.66	74.14	74.37	**74.43**
WiC	56.14	58.21	57.07	57.14	**58.79**	58.64	57.86	57.28	58.64	57.35	56.85	58.50	58.07	**58.14**
MNLI	71.19	77.83	**78.07**	74.77	77.26	73.29	77.91	70.80	77.25	77.91	74.31	**78.68**	72.59	78.23
QNLI	77.74	78.32	78.23	66.57	78.16	**78.80**	78.73	76.64	77.41	77.41	63.21	75.98	76.45	**78.01**
Avg.	69.67	71.62	71.58	67.37	72.06	71.18	**72.09**	69.78	71.34	71.65	66.26	71.83	70.37	**72.20**

Table 6: Comparison of 10-shot and 15-shot Results on different datasets. We have conducted additional experiments on the most challenging datasets (MRPC, WiC, MNLI, QNLI). These results confirm that Surprise Calibration (BSC) continues to outperform or match existing baselines under increased shot settings, maintaining robustness and effectiveness across harder tasks. Importantly, we also observed that performance improvements begin to saturate as the number of demonstrations increases beyond 5. This phenomenon is consistent with findings reported by Agarwal et al. (2024). For example, the improvement from 10-shot to 15-shot is often marginal (e.g., 72.09 → 72.20 in average accuracy). This trend suggests that SC is already effective in leveraging limited demonstration context, and adding more demonstrations yields only minor additional benefit, likely due to information redundancy in processing extended contexts.

DataSet	LM	Method						
		ICL	BC	LinC	CC+	BC+	LinC+	Ours
SST-2	Qwen2.5-3B	78.52	80.01	78.97	71.66	**81.16**	80.01	76.57
	Qwen2.5-7B	85.23	85.61	85.72	88.19	88.25	86.60	**91.32**
MNLI	Qwen2.5-3B	50.03	54.33	57.99	58.91	**70.01**	54.87	68.04
	Qwen2.5-7B	59.48	62.73	68.48	70.92	**77.78**	60.33	75.86
MRPC	Qwen2.5-3B	67.42	63.83	68.23	67.25	70.78	70.67	**72.34**
	Qwen2.5-7B	60.99	59.59	68.52	69.27	69.74	69.79	**71.19**
QNLI	Qwen2.5-3B	63.82	70.24	70.25	56.94	**74.42**	74.37	71.84
	Qwen2.5-7B	72.49	76.64	76.71	60.64	**77.21**	76.73	76.64
RTE	Qwen2.5-3B	64.62	67.51	67.51	62.81	72.20	**70.03**	70.04
	Qwen2.5-7B	69.31	71.84	72.20	71.84	74.37	74.73	**75.45**
WiC	Qwen2.5-3B	51.00	**55.50**	54.64	50.29	53.29	53.57	55.35
	Qwen2.5-7B	56.00	57.35	57.50	54.86	58.21	**58.50**	56.07
YouTube	Qwen2.5-3B	89.54	**89.80**	89.54	84.94	62.50	89.29	88.80
	Qwen2.5-7B	**89.79**	**89.79**	**89.79**	84.95	65.56	89.79	89.54
AIGA	Qwen2.5-3B	**72.82**	72.82	72.82	73.01	56.43	71.35	72.47
	Qwen2.5-7B	72.80	72.80	72.80	70.06	54.55	72.12	**73.42**
Avg.	Qwen2.5-3B	67.22	69.26	69.99	65.73	67.60	70.52	**71.93**
	Qwen2.5-7B	70.76	72.04	73.97	71.34	70.71	73.57	**76.18**

(a) 1-shot

DataSet	LM	Method						
		ICL	BC	LinC	CC+	BC+	LinC+	Ours
SST-2	Qwen2.5-3B	83.52	83.64	84.35	83.14	85.17	**85.83**	81.22
	Qwen2.5-7B	94.28	94.28	94.34	**94.67**	94.61	94.56	94.89
MNLI	Qwen2.5-3B	63.68	67.92	69.83	64.30	**77.96**	64.72	75.33
	Qwen2.5-7B	64.41	70.79	73.30	75.42	**82.58**	65.66	79.82
MRPC	Qwen2.5-3B	66.09	63.13	67.54	71.88	70.60	69.62	**72.17**
	Qwen2.5-7B	66.96	66.32	70.49	72.99	**74.43**	74.37	72.63
QNLI	Qwen2.5-3B	70.64	73.18	73.00	65.75	**78.47**	78.34	77.41
	Qwen2.5-7B	78.64	78.86	78.74	71.15	**80.57**	80.34	80.01
RTE	Qwen2.5-3B	73.65	76.53	76.53	71.12	75.09	75.45	**78.70**
	Qwen2.5-7B	80.87	81.23	80.87	75.09	77.25	79.06	**81.59**
WiC	Qwen2.5-3B	58.00	**58.35**	57.57	53.92	58.14	58.29	58.14
	Qwen2.5-7B	61.00	62.35	62.36	58.14	**63.07**	63.00	60.07
YouTube	Qwen2.5-3B	88.52	88.52	88.27	85.97	76.02	88.26	**88.78**
	Qwen2.5-7B	88.77	89.28	89.03	79.84	72.45	**91.32**	89.76
AIGA	Qwen2.5-3B	74.09	73.90	73.92	76.55	72.15	72.78	**74.39**
	Qwen2.5-7B	74.93	75.10	74.98	77.23	77.93	**79.65**	75.83
Avg.	Qwen2.5-3B	72.27	73.15	73.88	71.58	74.20	74.16	**75.76**
	Qwen2.5-7B	76.23	77.28	78.01	75.57	77.86	78.50	**79.33**

(b) 2-shot

DataSet	LM	Method						
		ICL	BC	LinC	CC+	BC+	LinC+	Ours
SST-2	Qwen2.5-3B	89.68	89.79	89.90	90.72	**91.27**	91.21	90.11
	Qwen2.5-7B	95.17	95.06	95.06	95.22	95.22	**95.27**	95.06
MNLI	Qwen2.5-3B	68.57	74.64	75.57	71.66	**78.86**	69.15	78.15
	Qwen2.5-7B	69.29	77.34	79.08	80.74	**84.05**	69.32	82.54
MRPC	Qwen2.5-3B	69.80	67.54	69.22	70.84	71.07	71.01	**71.82**
	Qwen2.5-7B	71.07	70.08	73.39	74.08	**74.67**	74.20	74.37
QNLI	Qwen2.5-3B	74.83	77.83	77.74	71.79	**79.64**	79.18	78.89
	Qwen2.5-7B	79.95	80.08	80.17	75.32	**81.18**	81.22	80.62
RTE	Qwen2.5-3B	77.98	85.56	85.19	76.53	78.34	78.34	**85.92**
	Qwen2.5-7B	84.11	84.11	**84.84**	78.70	83.03	84.83	84.11
WiC	Qwen2.5-3B	56.64	56.85	57.21	55.36	**58.93**	58.71	56.07
	Qwen2.5-7B	62.42	62.42	62.42	60.28	**63.85**	63.57	62.64
YouTube	Qwen2.5-3B	86.48	86.22	86.48	82.40	80.10	89.28	**90.05**
	Qwen2.5-7B	89.28	89.03	89.03	86.48	86.48	88.52	**90.82**
AIGA	Qwen2.5-3B	77.26	78.82	78.80	75.81	76.91	78.24	**79.43**
	Qwen2.5-7B	77.68	79.79	80.37	77.72	81.24	**82.50**	80.72
Avg.	Qwen2.5-3B	75.16	77.16	77.51	74.39	76.89	76.89	**78.81**
	Qwen2.5-7B	78.62	79.74	80.55	78.57	81.22	79.93	**81.36**

(c) 4-shot

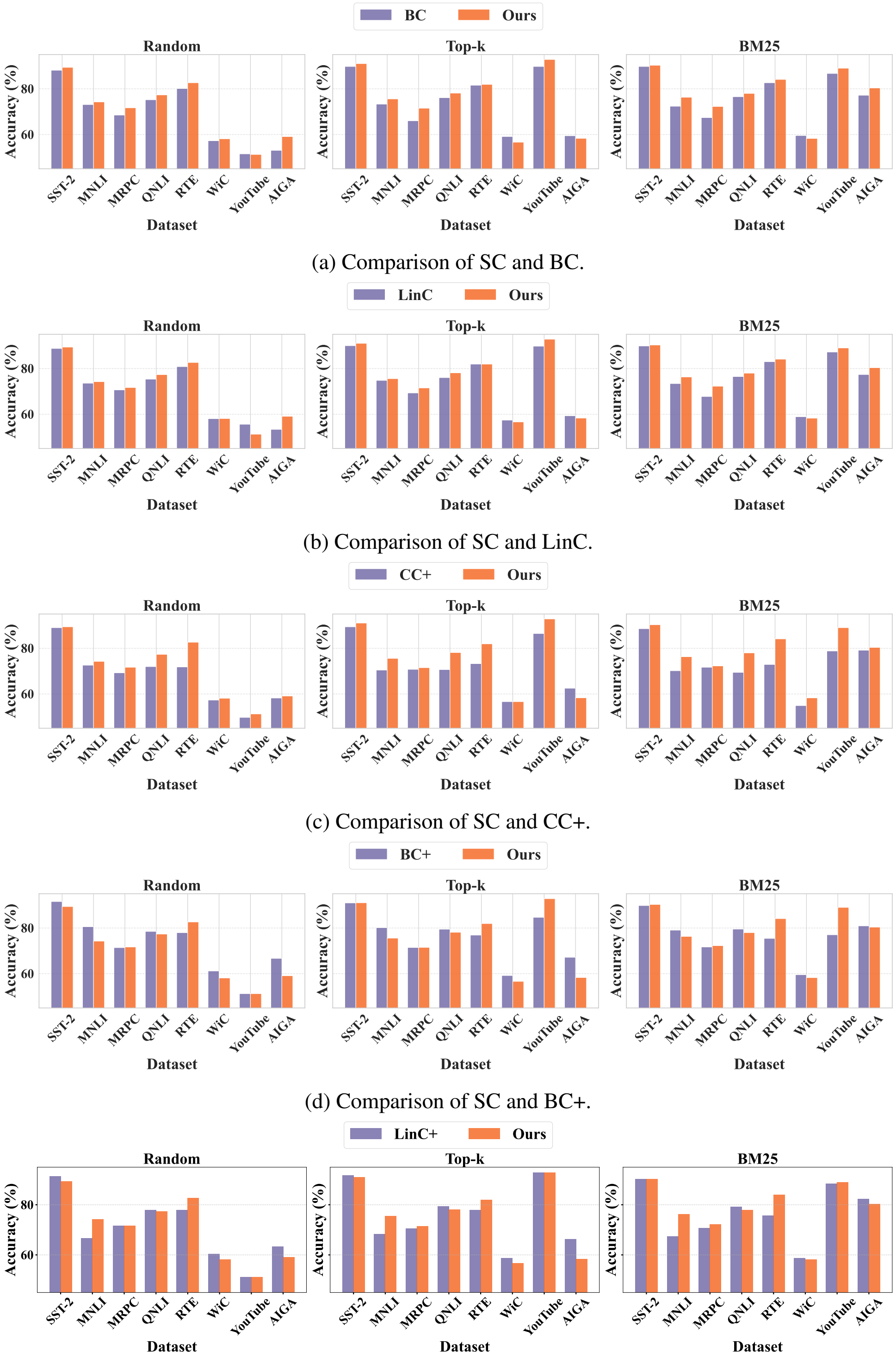
DataSet	LM	Method						
		ICL	BC	LinC	CC+	BC+	LinC+	Ours
SST-2	Qwen2.5-3B	90.22	90.28	90.44	91.49	**91.59**	91.37	90.44
	Qwen2.5-7B	94.12	94.17	94.34	94.23	94.34	94.34	**94.40**
MNLI	Qwen2.5-3B	69.25	75.87	76.55	72.65	78.63	70.42	**78.64**
	Qwen2.5-7B	70.33	78.19	79.03	82.13	**84.60**	71.40	83.26
MRPC	Qwen2.5-3B	71.30	69.39	70.31	70.72	71.30	71.07	**72.17**
	Qwen2.5-7B	72.00	70.26	72.99	73.45	**74.55**	73.57	73.85
QNLI	Qwen2.5-3B	76.20	77.81	77.81	71.05	79.17	**79.39**	78.47
	Qwen2.5-7B	79.90	79.71	79.79	73.09	**81.15**	81.11	**80.23**
RTE	Qwen2.5-3B	80.14	**85.20**	**85.20**	74.73	79.42	79.78	**85.20**
	Qwen2.5-7B	84.11	84.48	**84.84**	79.42	83.39	83.39	**84.48**
WiC	Qwen2.5-3B	57.50	57.35	57.28	56.42	**58.64**	58.14	57.71
	Qwen2.5-7B	62.64	62.07	62.50	62.43	**64.00**	63.28	63.71
YouTube	Qwen2.5-3B	84.94	88.26	88.26	87.86	79.59	87.76	**91.83**
	Qwen2.5-7B	87.76	89.80	89.54	87.50	86.73	90.31	**91.07**
AIGA	Qwen2.5-3B	78.90	79.88	80.01	74.86	77.81	79.29	**82.86**
	Qwen2.5-7B	79.50	81.86	82.01	77.05	80.75	82.74	**82.92**
Avg.	Qwen2.5-3B	76.06	78.01	78.23	74.97	77.02	77.15	**79.66**
	Qwen2.5-7B	78.80	80.07	80.63	78.66	81.19	80.02	**81.74**

(d) 5-shot

Table 7: Accuracy(%) comparison of different calibration methods on various datasets using BM25 selection strategy, increase ordering strategy, and Qwen2.5 models (3B and 7B) with 1- to 5-shot settings. The best performance for each dataset and model size is highlighted in bold.

DataSet	LM	Method						
		ICL	BC	LinC	CC+	BC+	LinC+	Ours
SST-2	Llama-3-8B	91.16	91.65	92.15	92.20	**93.52**	93.46	93.11 ± 0.03
MNLI	Llama-3-8B	53.26	53.89	54.37	60.26	**64.48**	59.02	61.22 ± 0.14
MRPC	Llama-3-8B	65.45	57.68	**67.94**	66.43	62.26	63.77	67.46 ± 0.21
QNLI	Llama-3-8B	61.52	62.35	62.38	54.34	**67.96**	66.70	67.01 ± 0.16
RTE	Llama-3-8B	66.42	66.42	67.51	65.70	69.68	68.23	**74.00 ± 0.00**
WiC	Llama-3-8B	55.50	55.50	55.29	54.35	**55.36**	55.29	54.41 ± 0.04
YouTube	Llama-3-8B	91.83	92.09	91.83	85.71	76.79	90.56	**92.34 ± 0.00**
AIGA	Llama-3-8B	80.41	81.97	**81.97**	81.61	81.01	**82.60**	81.50 ± 0.12
Avg.	Llama-3-8B	70.69	70.19	71.68	70.08	71.38	72.45	**73.88**

Table 8: Accuracy(%) comparison of different calibration methods on various datasets using BM25 selection strategy, increase ordering strategy, and Llama3-8B models with 3-shot settings. Results are reported as the mean ± standard deviation over three runs with three fixed random seeds.



(a) Comparison of SC and BC.

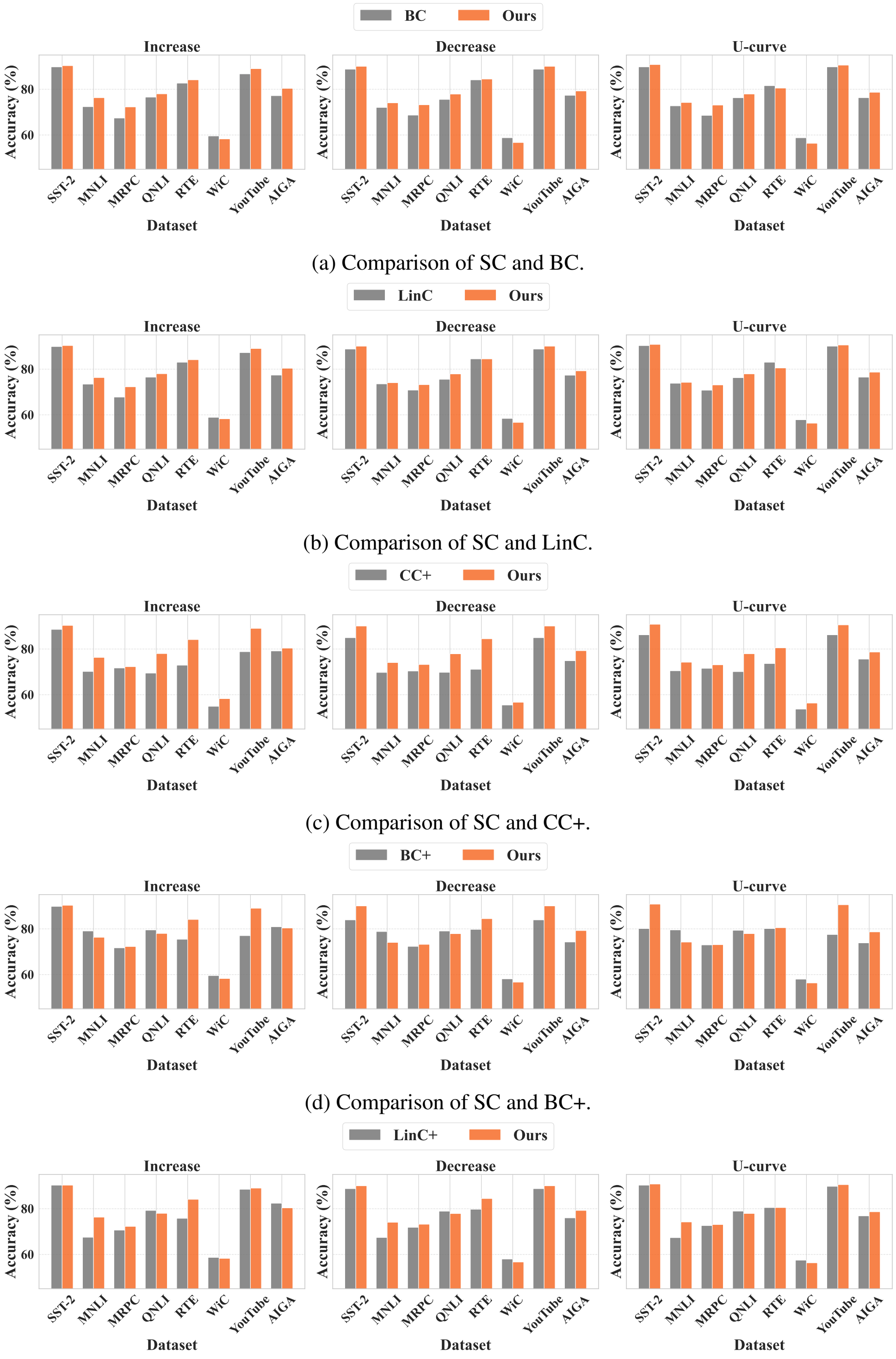
(b) Comparison of SC and LinC.

(c) Comparison of SC and CC+.

(d) Comparison of SC and BC+.

(e) Comparison of SC and LinC+.

Figure 10: Accuracy(%) comparisons between SC and various methods (BC, LinC, CC+, BC+, LinC+) across three demonstration selection strategies. Other settings are consistent with those shown in Table 2.



(a) Comparison of SC and BC.

(b) Comparison of SC and LinC.

(c) Comparison of SC and CC+.

(d) Comparison of SC and BC+.

(e) Comparison of SC and LinC+.

Figure 11: Accuracy(%) comparisons between SC and various methods (BC, LinC, CC+, BC+, LinC+) across three demonstration ordering strategies. Other settings remain consistent with those specified in Table 2.