CAN LARGE LANGUAGE MODELS ACHIEVE CALIBRATION WITH IN-CONTEXT LEARNING?

Chengzu Li[1], **Han Zhou**[1], **Goran Glavaš**[2], **Anna Korhonen**[1], **Ivan Vulić**[1]
[1]Language Technology Lab, University of Cambridge
[2]Center for Artificial Intelligence and Data Science, University of Würzburg
{cl917, hz416, iv250, alk23}@cam.ac.uk
goran.glavas@uni-wuerzburg.de

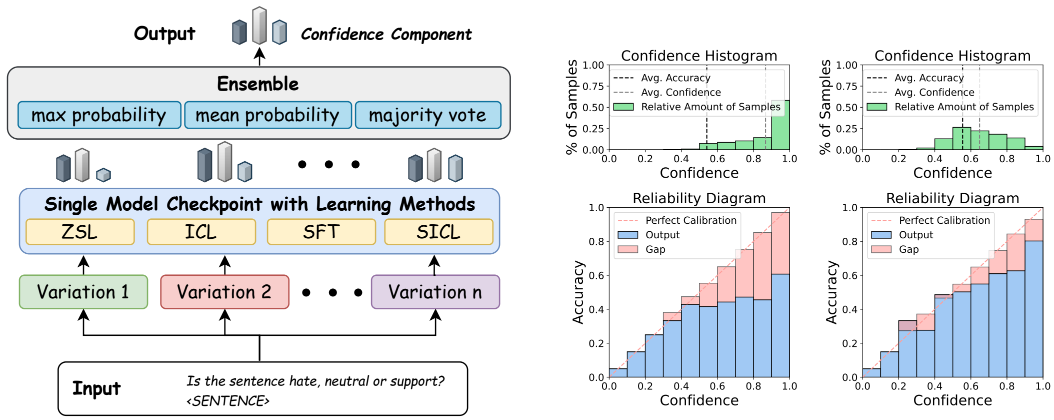
ABSTRACT

Following the standard supervised fine-tuning (SFT) paradigm, in-context learning (ICL) has become an efficient approach propelled by the recent advancements in large language models (LLMs), yielding promising performance across various tasks in few-shot data setups. However, both paradigms are prone to suffer from the critical problem of *overconfidence* (i.e., miscalibration), especially in such limited data setups. In this work, we deliver an in-depth analysis of the behavior across different choices of learning methods from the perspective of both performance and calibration, as well as their interplay. Through extensive controlled experiments, we find that simultaneous gains for both task performance and calibration are difficult to achieve, and the miscalibration problem exists across all learning methods in low-resource setups. To address this challenging trade-off between performance and calibration, we then study the potential of self-ensembling techniques applied at different modeling stages (e.g., variations of in-context examples or variations in prompts or different ensembling strategies). We justify the feasibility of self-ensembling on SFT in addition to ICL, to make the predictions more calibrated and have comparable or even better performance. Our work sheds light on which learning paradigm to choose and how to enhance both task performance and calibration of LLMs.

1 INTRODUCTION

Among different learning paradigms with large language models (LLMs) (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2023), Supervised Fine-Tuning (SFT) and In-Context Learning (ICL) have emerged as predominant methodologies (Raffel et al., 2020; Dong et al., 2022), demonstrating commendable efficacy across many tasks. Min et al. (2022) and Chen et al. (2022) introduce the in-context examples into training phrase, which we call supervised in-context learning (SICL). However, when the demonstrations, as a strong inductive bias, get combined with SFT, it has been shown that LLMs become more likely to fall into the problem of overconfidence (Desai & Durrett, 2020; Jiang et al., 2021) when specializing a (general-purpose) model; the predicted confidence distribution of ICL may be miscalibrated due to the bias in in-context examples (Fei et al., 2023). Through our extensive experiments in this work, we observe that both paradigms, SFT and ICL, suffer from the problem of miscalibration in low-resource scenarios.

The important challenges of overconfidence and miscalibration, particularly in scenarios marked by limited data availability, underscore the need for a nuanced understanding of these paradigms. However, most of the previous work (Mosbach et al., 2023; Sun et al., 2023) only focuses on comparing solely the performance of SFT and ICL on out-of-distribution (OOD) data, targeting general-purpose LLMs. Here, we instead focus on studying *task-specialized* language models, where the behavior of different paradigms' in-task performance along with their calibration remains an open research question. In addition, considering the possible issue of overconfidence and miscalibration, we pose and study another crucial research question: is it possible to ensure both in-task performance and well-calibrated LM behavior *at the same time*?



(a) Having the predictions with different variations to the input, we run self-ensembling on a single model to obtain final predictions and their confidence.

(b) The confidence histograms and reliability diagrams of SICL (left) and self-ensembled SICL with max probability (right) on SST5.

Figure 1: Illustration of the self-ensembled learning methods.

To address the above challenges, in this work we first investigate the performance and calibration of different model-tuning and ICL methods in limited data setups, based on which then explore the application of the self-ensembling method. Our contributions can be summarized as follows: **1)** We deliver a comprehensive empirical study with different choices of learning methods across a variety of tasks in limited data scenarios. **2)** We show the task-dependent relationship between in-task performance and calibration of LLMs and provide practical guidelines for the choice of learning paradigms (§3). **3)** We investigate and justify the feasibility of the self-ensembling method in enhancing both the performance and calibration of LLMs (§3). We release the code at: `https://github.com/cambridgeltl/ensembled-sicl`.

2 Methodology

In this paper, we analyze and compare four different learning paradigms in low-resource scenarios: zero-shot learning (ZSL),[1] in-context learning (ICL), supervised fine-tuning (SFT) and supervised in-context learning (SICL), depending on whether the input contains in-context examples and whether the model's parameters are fixed. With classification tasks in focus, we briefly describe each paradigm in Appendix A.1. For these learning paradigms, there are two points that create possible variations that can be used for ensembling: **1)** variation in the selection of in-context examples (for ICL and SICL), and **2)** variation in the chosen prompt (for all the paradigms). Previous work focuses on **1)** selecting a better combination of in-context examples (Su et al., 2023) for the model or **2)** generating an optimal prompting template (Zhou et al., 2023). On the other side, how the variation of multiple demonstration combinations and prompting templates influences the model behavior is still unexplored. Furthermore, we can **3)** 'self-ensemble' the model based on different ensembling strategies. We now introduce all these variants.

Variation of Ensembling Components. *a) Variation of In-Context Examples (Var-IC):* For ICL and SICL, IC examples and their ordering $[f_p(x_{IC}), y_{IC}]$ create variations in the model inputs with a fixed template f_p, while not impacting the test pair $f_p(x) \sim y$. This allows us to create various in-context example combinations as different inputs to a single model and obtain different ensemble components. *b) Variation of Prompting Templates (Var-Prompt):* Different prompting templates have shown high variance in task performance (Mishra et al., 2022). By changing the wording in the templates f_p, we can also create variations even with the same input to the model. For each input x, we randomly select a prompting template f'_p from a set of available prompting template candidates. In ICL and SICL, the same template is also applied to the in-context examples, formatting the final input as $[f'_p(x_1), y_1; ...; f'_p(x_M), y_M, f'_p(x)]$. This makes it applicable not only to ICL and SICL, but also to ZSL and SFT as well. *c) Variation of Both (Var-Both):* When we create a set of ensembling components, we can also combine these two variations.

[1]In the context of ZSL and later ICL there is no actual 'learning' taking the place, and the model simply reacts to the provided prompt, but we have chosen the term ZSL for consistency with previous literature.

Metrics	Methods	*SST-2*	*RTE*	*ANLI*	SST-5	NLU++	Manifestos	Hate Speech
Performance	**ZSL**	94.67	86.64	52.30	42.00	29.20	14.50	37.08
	ICL	$95.22_{0.12}$	88.45_{0}	$52.17_{0.47}$	$37.59_{0.23}$	$40.11_{0.09}$	$13.01_{0.19}$	$40.09_{0.08}$
	SFT	$95.61_{0.20}$	$\mathbf{88.81}_{0.29}$	$61.63_{1.68}$	$46.27_{2.09}$	$79.98_{0.59}$	$35.76_{1.23}$	$58.01_{1.01}$
	SICL	$\mathbf{95.63}_{0.29}$	$88.57_{0.45}$	$\mathbf{63.90}_{0.14}$	$\mathbf{47.12}_{1.93}$	$\mathbf{80.76}_{0.31}$	$\mathbf{37.55}_{1.61}$	$\mathbf{59.48}_{1.79}$
ECE	**ZSL**	**0.907**	**0.809**	0.356	**0.142**	0.231	0.432	0.318
	ICL	$0.915_{0.001}$	$0.815_{0.003}$	$0.351_{0.005}$	$0.183_{0.002}$	$0.129_{0.000}$	$0.476_{0.002}$	$0.271_{0.002}$
	SFT	$0.941_{0.011}$	$0.842_{0.002}$	$0.316_{0.023}$	$0.403_{0.032}$	$0.011_{0.001}$	$\mathbf{0.201}_{0.072}$	$0.354_{0.036}$
	SICL	$0.945_{0.013}$	$0.876_{0.003}$	$\mathbf{0.280}_{0.011}$	$0.360_{0.025}$	$\mathbf{0.002}_{0.001}$	$0.214_{0.038}$	$\mathbf{0.193}_{0.113}$

Table 1: Results for different learning methods across all 7 datasets. We report the average of 3 independent runs with different random seeds; variance is reported in the subscript. Numbers in **bold** represent the best performance and calibration score per dataset. The datasets 'seen' by Flan-T5 at pretraining are labeled in *italic*.

Self-Ensembling Strategy. For each variant, we obtain the predicted results $\hat{y}$ and the confidence $\hat{p}$ for each component. The next step involves ensembling the predictions over K different components. We experiment with three (self-) ensembling strategies to compare their impact on both performance and calibration.

Majority Vote: We select the predicted results that have the highest accumulated probability across K variants as the ensembling predictions. The accumulated probability $\mathcal{P}_{acc}$ for the predicted label l_i is defined as $\mathcal{P}_{acc}(\hat{y} = l_i) = \sum_{k=1}^{K} \mathcal{P}_k(\hat{y_k} = l_i)\mathbb{I}(\hat{y_k} = l_i)$. We pick the variants that have the same prediction as the ensembling prediction and average the probability distribution of the selected components $\mathcal{P}_{ens}(y|x) = \frac{1}{K'} \sum_{k=1}^{K'} \mathcal{P}_k(y|x)$ where K' is the number of selected variants.

Mean Probability: We average the predicted probability distribution of K variants and use the prediction that has the largest probability in the averaged distribution as the ensemble result $\hat{y_{ens}} = \arg\max_j \mathcal{P}_{ens}(y_j|x)$, where $\mathcal{P}_{ens}(y|x)$ is described as $\mathcal{P}_{ens}(y|x) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{P}_k(y|x)$.

Max Probability: For each possible value in the output space, we find the maximum probability of the predicted values across K variants and use this as the prediction's probability $\mathcal{P}'(\hat{y} = l_i|x) = \max(\mathcal{P}_j(\hat{y} = l_i|x), j \in [1, K])$. Because the probability is obtained from different components, the summation of these probabilities is not guaranteed to be 1. Therefore, we apply the normalization on the new probability distribution: $\mathcal{P}_{ens}(y|x) = \mathrm{Norm}(\mathcal{P}'(y|x))$. The ensemble prediction is determined as the $\hat{y}$ that has the highest probability after the ensembling step.

Estimating Calibration. Beyond task performance of all the possible variants, we estimate the calibration of the model's predictions (as a proxy towards model confidence) by using Expected Calibration Error (ECE) (Guo et al., 2017), the details of which are further introduced in Appendix A.5. We also report the negative log-likelihood (NLL) (Hastie et al., 2001) and information entropy (IE) as supplementary metrics of model's (lack of) confidence in Table 7 in Appendix B.1.

3 Results and Discussions

We consider 7 classification datasets that cover a range of label numbers and scenarios and use Flan-T5$_{\text{large}}$ (Chung et al., 2022) as the main model in the experiments. Detailed training setups are provided in Appendix A.3 and A.4, with prompting templates in Appendix D.1 and D.2.

Comparison between Learning Methods. In Table 1, we find that learning methods perform differently depending on the datasets and we divide the tasks into different families depending on their observed behavior. ICL demonstrates comparable performance to SFT/SICL on SST-2 and RTE, while tuning on these datasets with SFT/SICL yields increased ECE with no substantial in-task performance improvement. Conversely, tasks such as intent detection (NLU++), Manifestos, and Hate speech, show noticeable performance enhancement and better calibration by using SFT/SICL. We suspect the divergent behaviors are possibly due to data contamination of FLAN training corpus (Longpre et al., 2023) wherein ZSL and ICL exhibit similar performances with SFT and SICL on training datasets labeled as *seen* (e.g., SST-2, RTE).[2]

[2]We corroborate findings from other concurrent work on data contamination (Zhu et al., 2023; Deng et al., 2023) that also reports the unfair practice of evaluations on seen datasets.

Systems	Manifestos									
	Macro F1					ECE				
	Ori.	**Max**	**Mean**	**Majority**	**Δ**	**Ori.**	**Max**	**Mean**	**Majority**	**Δ**
ZSL	14.50				↑ 0.79	0.432				↓ 0.170
+ *Var-Prompt*	14.50	13.69	12.93	**15.29**	↑ 0.79	0.432	**0.262**	0.335	0.437	↓ 0.170
ICL	13.01				↑ 0.68	0.476				↓ 0.283
+ *Var-IC*	13.01	13.50	13.46	**13.69**	↑ 0.68	0.476	0.415	0.465	0.469	↓ 0.061
+ *Var-Prompt*	13.01	13.25	11.67	11.53	↑ 0.24	0.476	0.268	0.366	0.472	↓ 0.208
+ *Var-Both*	13.01	11.19	11.50	11.21	↓ 1.51	0.476	**0.193**	0.354	0.480	↓ 0.283
FT	35.76				↑ 0.73	0.201				↓ 0.134
+ *Var-Prompt*	35.39	**36.49**	35.66	34.91	↑ 1.10	0.144	**0.066**	0.105	0.135	↓ 0.077
SupICL	37.55				↑ 0.06	0.214				↓ 0.090
+ *Var-IC*	37.55	36.57	37.35	**37.61**	↑ 0.06	0.214	0.179	0.210	0.215	↓ 0.035
+ *Var-Prompt*	37.04	37.14	37.25	36.77	↑ 0.21	0.229	0.139	0.191	0.219	↓ 0.090
+ *Var-Both*	37.04	36.67	37.15	37.50	↑ 0.46	0.229	**0.124**	0.192	0.230	↓ 0.105

Table 2: Results of self-ensembling with different variations (selection). We mark the cells of baseline systems without self-ensembling in grey. Numbers in **bold** represents the best values for each learning method. Δ calculates the difference of performance and calibration error between the original results (Ori.) and the best self-ensembled results. We refer the readers to Appendix B.2 for full self-ensembling results.

Choice of Learning Methods. Given the comparison of the performances and calibration on different datasets, we suggest that the choice of learning methods should be task-dependent. The experiments and analysis indicate that unseen datasets obtain better performance and more trustworthy results with supervised tuning methods. For the seen datasets, ICL combined with other 'tweaks' such as model calibration can be a better choice, since the supervised tuning methods are more likely to make the model over-confident and less trustworthy. This is further justified that with Batch Calibration (Zhou et al., 2024) ICL performs on par or even better than SICL across all the (possibly) seen data but is still not comparable to those of either SICL or SFT for unseen datasets (Table 7 in Appendix B.1).

Nevertheless, despite these task-dependent variations, ECE remains relatively high across all methods except for intent detection, indicating the problem of *miscalibration across all learning methods*.

Self-Ensembling Results. Having observed the common miscalibration issues for all learning methods, we then investigate the feasibility of self-ensembling to improve calibration. In Table 2, with different learning methods combined with self-ensembling variations, we find that by changing the in-context example combinations or prompting templates, the best performance of self-ensembling outperforms the baseline without any ensembling by 0.79. Even though the performance gains seem marginal, self-ensembling substantially enhances the calibration performance, reducing the mean ECE value by 43%. We also notice that when self-ensembling over SFT and SICL, the model has lower ECE scores than ICL, but with much better task performance. This indicates the efficiency of self-ensembling in making the predictions more trustworthy while maintaining or even improving the task performance. It also suggests that *self-ensembling has the potential to mitigate the prominent problem of overconfidence in supervised tuning methods* (Figure 1b).

Different Variations and Ensembling Strategies. Our results suggest that with ICL, *Var-IC* yields more improvements than *Var-Prompt*, while the latter shows its efficacy with SFT and SICL. We also find that combining both variations may not necessarily improve the performance but is helpful in enhancing the trustworthiness. Regarding ensemble strategies, we notice that the majority vote improves the performance in general, but struggles to reduce the calibration error. Ensembling with max probability consistently produces the most faithful predictions with promising performances.

4 Conclusion

We have provided a comprehensive analysis of the intricate relationship between in-task performance and calibration across various learning methods in low-resource scenarios. Our findings illuminate the nuances of in-task performance and calibration across different task families, meanwhile addressing the inherent miscalibration over all learning methods. We have also investigated effective strategies to enhance both aspects, offering a viable solution through self-ensembling with more calibrated predictions and comparable or superior task performance. We hope that this study will contribute valuable insights into the dynamic landscape of LLMs. These discoveries also offer practical guidance to practitioners, aiding them in choosing suitable learning paradigms and paving the way for the development of more reliable and high-performing LLMs across diverse applications.

REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 719–730, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.53. URL `https://aclanthology.org/2022.acl-long.53`.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2022. URL `https://api.semanticscholar.org/CorpusID:247951931`.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. URL `https://doi.org/10.48550/arXiv.2210.11416`.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. *arXiv preprint arXiv:2311.09783*, 2023. URL `https://doi.org/10.48550/arXiv.2311.09783`.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 295–302, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.21. URL `https://aclanthology.org/2020.emnlp-main.21`.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. URL `https://doi.org/10.48550/arXiv.2301.00234`.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14014–14031, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.783. URL `https://aclanthology.org/2023.acl-long.783`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017. URL `http://proceedings.mlr.press/v70/guo17a.html`.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL `https://aclanthology.org/2021.tacl-1.57`.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: Designing data and methods for effective instruction tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22631–22648. PMLR, 2023. URL `https://proceedings.mlr.press/v202/longpre23a.html`.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.201. URL `https://aclanthology.org/2022.naacl-main.201`.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. Reframing instructional prompts to GPTk's language. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 589–612, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.50. URL `https://aclanthology.org/2022.findings-acl.50`.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12284–12314, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.779. URL `https://aclanthology.org/2023.findings-acl.779`.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL `https://api.semanticscholar.org/CorpusID:257532815`.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL `https://api.semanticscholar.org/CorpusID:160025533`.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Evgeniia Razumovskaia, Goran Glavaš, Anna Korhonen, and Ivan Vulić. Sqatin: Supervised instruction tuning meets question answering for improved dialogue nlu. *arXiv preprint arXiv:2311.09502*, 2023. URL `https://doi.org/10.48550/arXiv.2311.09502`.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/pdf?id=qY1hlv7gwg`.

Simeng Sun, Yang Liu, Dan Iter, Chenguang Zhu, and Mohit Iyyer. How does in-context learning help prompt tuning? *arXiv preprint arXiv:2302.11521*, 2023. URL `https://doi.org/10.48550/arXiv.2302.11521`.

Han Zhou, Xingchen Wan, Ivan Vulić, and Anna Korhonen. Survival of the most influential prompts: Efficient black-box prompt search via clustering and pruning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13064–13077, Singapore, December 2023. Association for Computational Linguistics. URL `https://aclanthology.org/2023.findings-emnlp.870`.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=L3FHMoKZcS`.

Wenhong Zhu, Hongkun Hao, Zhiwei He, Yunze Song, Yumeng Zhang, Hanxu Hu, Yiran Wei, Rui Wang, and Hongyuan Lu. Clean-eval: Clean evaluation on contaminated large language models. *arXiv preprint arXiv:2311.09154*, 2023. URL `https://doi.org/10.48550/arXiv.2311.09154`.

A Experiment Setup

A.1 Learning Methods Definition

Zero-Shot Learning (ZSL). Given the input x and the prompting template f_p, the prediction $\hat{y}$ from the LM can be represented as $\hat{y} = \arg\max_j \mathcal{P}(y_j|f_p(x))$, where the parameters of the underlying LM are fixed. The prompting template $f_p(x)$ includes the task instructions and special symbols which can be replaced by the input x. We attach the prompting templates for different classification tasks in Appendix D.

In-Context Learning (ICL). Similar to ZSL, instead of only feeding the input x to the model, we first prepend M in-context examples (IC) (also called demonstrations) $[f_p(x_1), y_1; ...; f_p(x_M), y_M]$ to the input x. The examples are retrieved from the pool of examples R following (random or non-random) selection strategy. The prediction is then defined as $\hat{y} = \arg\max_j \mathcal{P}(y_j|[f_p(x_{IC}), y_{IC}], f_p(x))$.

Supervised Fine-Tuning (SFT). As mentioned, ZSL and ICL are inference-only paradigms treating the LM as a black box. On the other hand, FT first trains the model on the training set following the input format $f_p(x)$ from ZSL. Note that here we use SFT to refer to instruction-style fine-tuning with a prompting template. Inference with the tuned model $\mathcal{P}'$ is then conducted in the same way as with ZSL. Both during training and inference, we can use different prompting templates to create variations in the model input, which we further elaborate on in §2.

Supervised In-Context Learning (SICL). Based on the propositions from Min et al. (2022) and Chen et al. (2022), we can also fine-tune the model to directly optimize the in-context learning objective. For each training step, M in-context examples $(x_1, y_1), ..., (x_M, y_M)$ are selected from the pool R. We then prepend the selected in-context examples to the input x as before with ICL and use the concatenation as the final model input, and train the model to generate y. Inference proceeds in the same way as with ICL, except that we now use the task-tuned model $\mathcal{P}'$.

A.2 Dataset Details

SST-2. The SST-2 dataset, a widely-used benchmark in sentiment analysis, comprises sentences from movie reviews annotated with binary sentiment labels (positive or negative). We train the model with the data randomly sampled from the original training set and report the performance on the test set. We evaluate the model's performance based on accuracy.

SST-5. SST-5, an extension of SST-2, enhances sentiment analysis with five classes: very negative, negative, neutral, positive, and very positive. Derived from movie reviews, this dataset provides a nuanced perspective on sentiment, allowing models to distinguish fine-grained emotional tones. With all other practices aligned with SST-2, the results are evaluated with micro f1 and macro f1 scores because it has more than 2 labels.

RTE. Recognizing Textual Entailment is a benchmark dataset assessing the task of determining logical entailment between pairs of text snippets. Annotated with binary labels indicating entailment or not, RTE is crucial for evaluating models' logical reasoning abilities. We report the accuracy in accordance with other binary classification tasks.

ANLI. Adversarial NLI is a benchmark dataset introducing adversarial examples to challenge models with nuanced reasoning and complex inferences. With labeled sentence pairs denoting entailment, contradiction, or neutrality, ANLI is crucial for assessing models' robustness and generalization in the face of diverse linguistic challenges. ANLI has three different rounds of contexts, with later rounds having a better base model, thus being more difficult for the model to distinguish. In this paper, we conduct the experiments mainly on the first round, which is easier than other rounds, in order to compare the performance with ICL. Since it is a multiclass classification task, we report the performance with micro and macro F1 scores. In this paper, we mainly use r1 level data for experiments.

Dataset	Label number	Main Metric	Train size
SST-2	2	acc	50
RTE	2	acc	50
ANLI	3	acc	50
SST5	5	macro f1	50
NLU++	2	micro f1	50
Manifestos	8	macro f1	800
Hate speech	3	macro f1	50

Table 3: Summary of the datasets, main evaluation metric for performance and training data size used for experiment.

Task	Label Verbalizer
SST2	postive, negative
RTE	yes, no
ANLI	yes, maybe, no
SST5	terrible, bad, neutral, good, great
NLU++	yes,no
Manifestos	other, external, democracy, political, economy, welfare, fabric, group
Hate Speech	support, neutral, hate

Table 4: Label verbalizer for different tasks.

NLU++. NLU++ is a more challenging benchmark in task-oriented dialogue system with more fine-grained domain ontologies and sentences with multiple intents. It has two tasks: intent detection and slot labeling, covering the *banking* and *hotels* two domains. In this work, we focus on the intent detection task, which is a multi-label classification task and we follow the setting from recent work with state-of-the-art results (Razumovskaia et al., 2023), which formats it as a binary `yes/no` classification task. See the cited work for further details. Regarding the data split, 1,000 sentences from NLU++ were held out for testing and 50 sentences from the leftover 2k+ sentences were sub-sampled for training.

Manifestos. Manifestos was originally created to collect the manifestos of parties from different countries. It also includes the analytical variables that indicates the respective categories of the quasi-sentences. The corpus have 8 domains overall, which are listed as follows: None (of the below) / Other, External Relations, Freedom and Democracy, Political System, Economy, Welfare and Quality of Life, Fabric of Society, Social Groups. In this paper, we use the sentences that only have one golden domain and exclude the ones with multiple labels.

Measuring Hate Speech Corpus. Measuring Hate Speech Corpus, in short Hate speech, contains1 10 constituent ordinal labels and the continuous hate speech score to measure the extent of hate. We use the hate speech score as indicator of hate speech in this paper. We follow the original division of approximate hate speech provided by the authors, where ¿ 0.5 is approximately hate speech, ¡ -1 is counter or supportive speech, and -1 to 0.5 is neutral or ambiguous.

We only experiment on the intent detection task in the NLU++ bank domain and for ANLI we mainly discuss r1 level data. We summarize the training data size, main performance evaluation metrics, and the number of labels for each dataset in Table 3. We also list the label verbalizers for all datasets in Table 4.

A.3 Environment Setup

We mainly use Flan-$T5_{large}$ (783M parameters) as the task models for all the datasets. We also use Flan-$T5_{xl}$ (2.85B parameters) on some of the task to see whether the findings still hold on the larger model. For SFT and SICL, we use LoRA (Hu et al., 2022) to tune Flan-$T5_{xl}$. Due to the computational limitations, we can't obtain the results on all the datasets with Flan-$T5_{xl}$.

All the experiments are conducted on Cambridge High-Performance Clusters with a single A100 (80G) and a 32-core vCPU. We release the code and the environment dependencies for reproducible purposes at `[URL-ANONYMOUS]`.

A.4 Hyperparameters

In order to evaluate the model's performance and trustworthiness in low-resource scenarios, we sample a subset of the training set and evaluate it on a fixed set of data as an evaluation and test set. For Manifestos, because it has 8 classes and is more expertise in specialized domains (politics, economics and etc.), we use a relatively larger training set to adapt the model to the task itself. For Hate Speech, we manually sample the training set and test set ourselves since the corpus didn't provide the split. We randomly sample 1500 data as the fixed test set and 500 examples as the fixed evaluation set.

All the main experiments are conducted three times with 0, 21, 42 as the random seeds. We report the mean values of three runs in the main content.

Across different learning paradigms (ICL and SICL), we concatenate 3 in-context examples in front of the input for the main experiments.

For supervised fine-tuning methods, we attach the detailed hyperparameters in Table 5 for reproducibility. Because tuning the model in the low-resource setting is prone to over-confidence, in order to mitigate the problem, we apply the early stopping with the patience of 5.

Regarding the configuration hyper-parameters of PEFT, they are listed in Table 6. Unlisted properties use the default values in PEFT implementation from huggingface[3].

A.5 Calibration Measurement

Expected Calibration Error (ECE) (Guo et al., 2017) divides the n predicted results based on their confidence into M bins B_1 to B_M and then computes a weighted average over the absolute difference between the accuracy $\mathrm{acc}(B_m)$ and mean confidence $\mathrm{conf}(B_m)$ of the predictions within each bin. We set M to 10 in this work.

$$\widehat{\mathrm{ECE}} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)| \tag{1}$$

ECE measures the difference between the model's empirical accuracy and its confidence (predicted probability). The smaller the ECE, the more confident the model prediction would be.

B Full Experiment Results

To solidify the empirical findings, in this section, we present full experiment results with more metrics in addition to the table in the main content for readers' reference.

B.1 Results of different learning methods

Table 7 shows the full results on all 7 datasets. We report the accuracy, micro f1, and macro f1 as the performance metrics. We report ECE as the measurement of calibration. We also include NLL and IE as supplementary uncertainty metrics. We find that on SST-2 and RTE, the model achieves comparable or even better performance with ZSL and ICL than SFT and SICL. In the meantime, the predictions have a relatively high ECE, indicating that on these two datasets, the model has the issue of miscalibration. On ANLI, Manifestos, Hate speech, and NLU++, with SFT and SICL the model has lower calibration error than ICL and achieves better performance in both performance metrics.

In addition to the original results, we include the Batch Calibration results across all the datasets, as shown in Table 7. On SST-5 and ANLI, although ZSL and ICL achieve similar micro f1 scores,

[3] `https://huggingface.co/docs/peft/index`

Hyperparameters	**ICL**	**FT**	**SupICL**
	SST2		
train batch size	-	8	8
eval batch size	64	32	32
grad accumulation	-	1	1
learning rate	-	5e-5	5e-5
evaluation per steps	-	10	10
max training epochs	-	200	200
early stopping patience	-	5	5
early stopping metric	-	accuracy	accuracy
	RTE		
train batch size	-	8	4
eval batch size	64	32	32
grad accumulation	-	1	2
learning rate	-	5e-5	5e-5
evaluation per steps	-	10	10
max training epochs	-	200	200
early stopping patience	-	5	5
early stopping metric	-	accuracy	accuracy
	ANLI		
train batch size	-	8	4
eval batch size	64	32	32
grad accumulation	-	1	2
learning rate	-	5e-5	5e-5
evaluation per steps	-	10	10
max training epochs	-	200	200
early stopping patience	-	5	5
early stopping metric	-	accuracy	accuracy
	SST5		
train batch size	-	8	8
eval batch size	64	32	32
grad accumulation	-	1	1
learning rate	-	5e-5	5e-5
evaluation per steps	-	10	10
max training epochs	-	200	200
early stopping patience	-	5	5
early stopping metric	-	macro f1	macro f1
	NLU++		
train batch size	-	16	16
eval batch size	64	32	32
grad accumulation	-	2	2
learning rate	-	5e-5	5e-5
evaluation per steps	-	500	500
max training epochs	-	200	200
early stopping patience	-	5	5
early stopping metric	-	micro f1	micro f1
	Manifestos		
train batch size	-	8	4
eval batch size	32	32	32
grad accumulation	-	1	2
learning rate	-	5e-5	5e-5
evaluation per steps	-	10	10
max training epochs	-	200	200
early stopping patience	-	5	5
early stopping metric	-	macro f1	macro f1
	Hate speech		
train batch size	-	8	4
eval batch size	32	32	32
grad accumulation	-	1	2
learning rate	-	5e-5	5e-5
evaluation per steps	-	10	10
max training epochs	-	200	200
early stopping patience	-	5	5
early stopping metric	-	macro f1	macro f1

Table 5: Hyper-parameters for each dataset when comparing different learning methods.

Hyperparameters	Values
r	8
lora alpha	32
lora dropout	0.05
target modules	q, v

Table 6: Hyper-parameters for PEFT with FlanT5-xl.

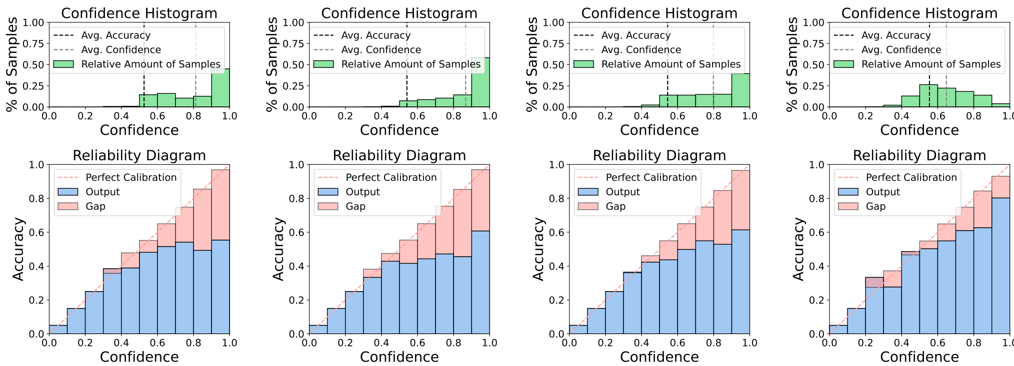
Evaluation Metrics		SST2				RTE			
		ZSL	ICL	SFT	SICL	ZSL	ICL	SFT	SICL
Performance	**acc**	94.67	$95.22_{0.12}$	$95.61_{0.20}$	$95.63_{0.29}$	86.64	88.45_{0}	$88.81_{0.29}$	$88.57_{0.45}$
	macro f1	-	-	-	-	-	-	-	-
+ calibrated	**acc**	95.50	$95.95_{0.19}$	$95.61_{0.20}$	$95.63_{0.29}$	89.53	$90.25_{0.51}$	$89.17_{0.29}$	$87.97_{0.45}$
	macro f1	-	-	-	-	-	-	-	-
Trustworthiness	**ECE**	0.9069	$0.9149_{0.0014}$	$0.9408_{0.0113}$	$0.9449_{0.0129}$	0.8092	$0.8150_{0.0030}$	$0.8418_{0.0224}$	$0.8759_{0.0025}$
	NLL	0.1465	$0.1347_{0.0006}$	$0.2146_{0.0962}$	$0.2710_{0.1396}$	0.3438	$0.2910_{0.0037}$	$0.4421_{0.2133}$	$1.1234_{0.1133}$
	IE	0.0615	$0.0599_{0.0003}$	$0.0215_{0.0149}$	$0.0154_{0.0124}$	0.0977	$0.0975_{0.0002}$	$0.0630_{0.0304}$	$0.0153_{0.0030}$
+ calibrated	**ECE**	0.7877	$0.7961_{0.0016}$	$0.8040_{0.0056}$	$0.8064_{0.0096}$	0.6831	$0.6991_{0.0054}$	$0.7022_{0.0061}$	$0.7111_{0.0045}$
Evaluation Metrics		**SST5**				**ANLI**			
		ZSL	ICL	SFT	SICL	ZSL	ICL	SFT	SICL
Performance	**micro f1**	52.58	$50.48_{0.15}$	$50.59_{1.38}$	$54.25_{0.46}$	52.30	$52.17_{0.47}$	$61.63_{1.68}$	$63.90_{0.14}$
	macro f1	42.00	$37.59_{0.23}$	$46.27_{2.09}$	$47.12_{0.0193}$	42.07	$42.06_{0.39}$	$61.17_{1.76}$	$63.53_{0.43}$
+ calibrated	**micro f1**	50.05	$50.89_{0.86}$	$50.35_{0.50}$	$49.91_{0.21}$	62.30	$61.27_{0.82}$	$62.47_{1.60}$	$64.50_{0.41}$
	macro f1	48.98	$49.80_{0.91}$	$50.43_{0.68}$	$49.89_{0.26}$	61.98	$61.15_{0.83}$	$62.36_{1.67}$	$64.46_{0.40}$
Trustworthiness	**ECE**	0.1416	$0.1833_{0.0020}$	$0.4030_{0.0321}$	$0.3602_{0.0250}$	0.3555	$0.3511_{0.0050}$	$0.3161_{0.0230}$	$0.2803_{0.0108}$
	NLL	1.1762	$1.2261_{0.0021}$	$3.1986_{1.4010}$	$2.3390_{0.2827}$	4.7484	$3.8961_{0.0233}$	$2.2599_{0.3255}$	$1.9226_{0.2912}$
	IE	0.1599	$0.1522_{0.0001}$	$0.0454_{0.0209}$	$0.0489_{0.0107}$	0.0945	$0.0987_{0.0001}$	$0.0587_{0.0070}$	$0.0663_{0.0113}$
+ calibrated	**ECE**	0.1010	$0.1027_{0.0084}$	$0.0720_{0.0129}$	$0.0456_{0.0035}$	0.0709	$0.0430_{0.0079}$	$0.0626_{0.0145}$	$0.0495_{0.0085}$
Evaluation Metrics		**Manifestos**				**Hate speech**			
		ZSL	ICL	SFT	SICL	ZSL	ICL	SFT	SICL
Performance	**micro f1**	20.87	$19.29_{0.16}$	$37.54_{1.10}$	$38.12_{2.01}$	39.67	$40.18_{0.14}$	$59.67_{0.47}$	$61.33_{2.05}$
	macro f1	14.50	$13.01_{0.19}$	$35.76_{1.23}$	$37.55_{1.61}$	37.08	$40.09_{0.08}$	$58.01_{1.01}$	$59.48_{1.79}$
+ calibrated	**micro f1**	33.63	$31.00_{0.37}$	$38.58_{0.46}$	$38.33_{0.33}$	43.87	$45.11_{0.46}$	$59.89_{0.17}$	$59.58_{3.26}$
	macro f1	30.86	$29.15_{0.53}$	$37.57_{1.05}$	$37.83_{0.41}$	40.86	$42.36_{0.42}$	$58.06_{0.27}$	$57.81_{3.03}$
Trustworthiness	**ECE**	0.4319	$0.4760_{0.0018}$	$0.2005_{0.0723}$	$0.2138_{0.0376}$	0.3175	$0.2708_{0.0024}$	$0.3541_{0.0364}$	$0.1928_{0.1131}$
	NLL	3.7344	$3.9423_{0.0091}$	$2.0725_{0.1637}$	$2.0264_{0.0751}$	1.3836	$1.2136_{0.0057}$	$4.4720_{2.4332}$	$2.0827_{1.6186}$
	IE	0.1141	$0.1006_{0.0001}$	$0.1469_{0.0182}$	$0.1446_{0.0104}$	1.0112	$0.2426_{0.0007}$	$0.0391_{0.0286}$	$0.1447_{0.0963}$
+ calibrated	**ECE**	0.0356	$0.0721_{0.0020}$	$0.0487_{0.0120}$	$0.0654_{0.0043}$	0.1107	$0.1173_{0.0045}$	$0.0646_{0.0274}$	$0.0453_{0.0114}$
Evaluation Metrics		**NLU++**							
		ZSL	ICL	SFT	SICL				
Performance	**micro f1**	29.2	$40.11_{0.09}$	$79.98_{0.59}$	$80.76_{0.31}$				
	macro f1	40.26	$51.96_{0.04}$	$80.58_{1.00}$	$80.49_{0.12}$				
+ calibrated	**micro f1**	11.18	$12.25_{0.02}$	$16.45_{0.72}$	$21.00_{4.78}$				
	macro f1	11.35	$12.56_{0.05}$	$21.27_{2.05}$	$27.07_{3.58}$				
Trustworthiness	**ECE**	0.2311	$0.1291_{0.0001}$	$0.0112_{0.0007}$	$0.0020_{0.0007}$				
	NLL	0.3435	$0.2140_{0.0001}$	$0.1305_{0.0348}$	$0.0839_{0.0127}$				
	IE	0.2268	$0.1419_{0.0001}$	$0.0014_{0.0007}$	$0.0020_{0.0007}$				
+ calibrated	**ECE**	0.4679	$0.4559_{0.0001}$	$0.4183_{0.0038}$	$0.4098_{0.0049}$				

Table 7: Full experiment results across 7 datasets with different learning methods. We report the mean value for 3 runs with different random seeds and list the variance in the subscripts. We color the Batch Calibration results in grey.

there is still a gap in the original macro f1 scores between ZSL/ICL and SFT/SICL. However, after applying Batch Calibration, we find that ZSL/ICL has a comparable macro f1 score to SFT/SICL. On Manifestos, Hate speech, and NLU++, we don't observe comparable performance between ZSL/ICL and SFT/SICL either with or without Batch Calibration.

B.2 Results of self-ensembling

Table 8 shows the self-ensembling results across 4 datasets. We exclude the seen datasets (SST-2 and RTE) for fair evaluation, as well as NLU++ since it's almost well-calibrated with supervised tuning. We still include ANLI for comparison even though it is included during pre-training. We report the mean values of the results with 3 different random seeds (0, 21, 42).



(a) FT (b) SICL (c) E-FT(*Var-Prompt*) (d) E-SICL(*Var-Both*)

Figure 2: The confidence histograms and reliability diagrams of SFT and SICL on SST-5 with or without self-ensembling using max probability.

From the perspective of performance, we find that on SST-5, Manifestos, and Hate speech, self-ensembled results achieve slightly better performances on average and show positive improvements with each learning method. On ANLI, we observe no significant improvement in the accuracy of self-ensembled results and the decreases in performance are trivial as well. However, from the perspective of calibration, we find that self-ensembling with max probability consistently decreases the calibration over all settings, as shown in Figure 2. Introducing variations in both in-context examples and prompting templates yields the lowest calibration error in all experiments.

Among different ensembling methods, we find that majority vote can achieve better performances sometimes but it doesn't help to reduce the calibration error or even make it worse. Mean probability and max probability are able to improve the performance meanwhile reducing the calibration error. The empirical experiment results suggest that although majority vote as a widely used ensemble method achieves better performance, it is worth noting that it may deliver unfaithful predictions, which is not preferred in real application.

C Supplementary Results for Ablations

C.1 How about larger models?

Table 9 shows the results on SST-5 and Hate speech with different learning methods using Flan-T5$_{\text{xl}}$. With ZSL and ICL, we observe that xl version model has larger calibration errors than Flan-T5$_{\text{large}}$ model on possibly seen datasets (SST-2 and SST-5), whereas on unseen datasets (Hate speech and Manifestos) it shows lower ECE. Regarding the performances, the xl model shows better performances on unseen datasets than the large version model but doesn't guarantee better performances on seen datasets. After tuning the model with SFT or SICL, we find that the calibration errors are reduced across all tasks, which is different from Flan-T5$_{\text{large}}$. Due to the computation constraint, we leave the discrepancy in the behaviors of different-sized models to future work.

Table 10 shows the self-ensembled results using Flan-T5$_{\text{xl}}$ on SST-5 and Hate speech. We find that both the performances and calibration errors get better with self-ensembling, justifying the feasibility and extensibility of self-ensembling on larger models. Compared with Flan-T5$_{\text{large}}$, the self-ensembled xl model yields much lower calibration errors with SFT and SICL on both tasks.

We also surprisingly find that the self-ensembling method can improve both the performance and calibration on the task that Batch Calibration finds struggling. On Hate speech, after applying Batch Calibration, we witnessed an improvement in calibration along with a drop in performance. However, when we apply self-ensembling, the predictions yield better performance and much lower calibration errors at the same time. This indicates the potential of self-ensembled language models in producing better and more reliable predictions.

Systems	SST5									
	Macro F1					ECE				
	Ori.	**Max**	**Mean**	**Majority**	**Δ**	**Ori.**	**Max**	**Mean**	**Majority**	**Δ**
ZSL	42.00				↑ 4.79	0.1416				↓ 0.0634
+ *Var-Prompt*	42.00	**46.79**	45.70	44.82	↑ 4.79	0.1416	**0.0782**	0.1132	0.1374	↓ 0.0634
ICL	37.59				↑ 0.16	0.1833				↓ 0.0911
+ *Var-IC*	37.59	**37.75**	37.33	37.26	↑ 0.16	0.1833	0.1198	0.1749	0.1857	↓ 0.0635
+ *Var-Prompt*	37.59	37.13	36.52	36.83	↓ 0.46	0.1833	0.1363	0.1838	0.2025	↓ 0.0470
+ *Var-Both*	37.59	33.82	35.33	35.78	↓ 1.81	0.1833	**0.0955**	0.1832	0.2107	↓ 0.0911
FT	46.27				↑ 2.08	0.4030				↓ 0.1022
+ *Var-Prompt*	48.11	47.43	**48.35**	48.33	↑ 0.24	0.3960	**0.3008**	0.3466	0.3973	↓ 0.0952
SupICL	47.12				↑ 0.79	0.3602				↓ 0.2402
+ *Var-IC*	47.12	47.30	47.37	47.31	↑ 0.25	0.3602	0.2755	0.3428	0.3615	↓ 0.0847
+ *Var-Prompt*	47.99	47.70	47.88	**47.91**	↓ 0.08	0.2714	0.1698	0.2342	0.2801	↓ 0.1016
+ *Var-Both*	47.99	47.18	47.56	47.54	↓ 0.43	0.2714	**0.1200**	0.2296	0.2791	↓ 0.1514

Systems	Manifestos									
	Macro F1					ECE				
	Ori.	**Max**	**Mean**	**Majority**	**Δ**	**Ori.**	**Max**	**Mean**	**Majority**	**Δ**
ZSL	14.50				↑ 0.79	0.4319				↓ 0.1699
+ *Var-Prompt*	14.50	13.69	12.93	**15.29**	↑ 0.79	0.4319	**0.2620**	0.3349	0.4374	↓ 0.1699
ICL	13.01				↑ 0.68	0.4760				↓ 0.2828
+ *Var-IC*	13.01	13.50	13.46	**13.69**	↑ 0.68	0.4760	0.4146	0.4645	0.4689	↓ 0.0614
+ *Var-Prompt*	13.01	13.25	11.67	11.53	↑ 0.24	0.4760	0.2682	0.3661	0.4718	↓ 0.2078
+ *Var-Both*	13.01	11.19	11.50	11.21	↓ 1.51	0.4760	**0.1932**	0.3537	0.4796	↓ 0.2828
FT	35.76				↑ 0.73	0.2005				↓ 0.1343
+ *Var-Prompt*	35.39	**36.49**	35.66	34.91	↑ 1.10	0.1435	**0.0662**	0.1049	0.1352	↓ 0.0773
SupICL	37.55				↑ 0.06	0.2138				↓ 0.0902
+ *Var-IC*	37.55	36.57	37.35	**37.61**	↑ 0.06	0.2138	0.1789	0.2096	0.2152	↓ 0.0349
+ *Var-Prompt*	37.04	37.14	37.25	36.77	↑ 0.21	0.2285	0.1388	0.1912	0.2190	↓ 0.0897
+ *Var-Both*	37.04	36.67	37.15	37.50	↑ 0.46	0.2285	**0.1236**	0.1918	0.2303	↓ 0.1049

Systems	Hate Speech									
	Macro F1					ECE				
	Ori.	**Max**	**Mean**	**Majority**	**Δ**	**Ori.**	**Max**	**Mean**	**Majority**	**Δ**
ZSL	37.08				↓ 0.13	0.3175				↓ 0.0489
+ *Var-Prompt*	37.08	36.54	**36.95**	**36.95**	↓ 0.13	0.3175	**0.2686**	0.3024	0.3200	↓ 0.0489
ICL	40.09				↑ 1.10	0.2708				↓ 0.1112
+ *Var-IC*	40.09	40.01	39.98	40.49	↑ 0.40	0.2708	0.2332	0.2668	0.2694	↓ 0.0376
+ *Var-Prompt*	40.09	41.03	**41.19**	41.05	↑ 1.10	0.2708	0.1944	0.2359	0.2745	↓ 0.0764
+ *Var-Both*	40.09	39.68	40.30	40.49	↑ 0.40	0.2708	**0.1596**	0.2366	0.2776	↓ 0.1112
FT	58.01				↓ 0.82	0.3541				↓ 0.1155
+ *Var-Prompt*	55.92	57.16	57.17	**57.19**	↑ 1.27	0.3502	**0.2386**	0.2899	0.3448	↓ 0.1116
SupICL	59.48				↑ 0.74	0.1928				↓ 0.0777
+ *Var-IC*	59.48	59.98	59.82	59.83	↑ 0.50	0.1928	0.1413	0.1789	0.1905	↓ 0.0515
+ *Var-Prompt*	58.66	59.96	60.10	59.86	↑ 1.44	0.2507	0.1648	0.2111	0.2457	↓ 0.0859
+ *Var-Both*	58.66	60.12	**60.22**	59.97	↑ 1.56	0.2507	**0.1151**	0.2057	0.2455	↓ 0.1356

Systems	ANLI									
	Acc					ECE				
	Ori.	**Max**	**Mean**	**Majority**	**Δ**	**Ori.**	**Max**	**Mean**	**Majority**	**Δ**
ZSL	52.30				↓ 0.60	0.3555				↓ 0.0133
+ *Var-Prompt*	52.30	51.70	51.60	51.60	↓ 0.60	0.3555	**0.3422**	0.3603	0.3640	↓ 0.0133
ICL	52.17				↓ 0.01	0.3511				↓ 0.0204
+ *Var-IC*	52.17	52.10	52.10	52.10	↓ 0.07	0.3511	0.3347	0.3506	0.3521	↓ 0.0164
+ *Var-Prompt*	52.17	52.03	52.07	**52.16**	↓ 0.01	0.3511	0.3375	0.3490	0.3506	↓ 0.0136
+ *Var-Both*	52.17	**51.70**	51.80	51.80	↓ 0.37	0.3511	**0.3307**	0.3510	0.3544	↓ 0.0204
FT	61.63				↓ 0.07	0.3161				↓ 0.0271
+ *Var-Prompt*	61.53	**61.56**	61.37	61.37	↑ 0.03	0.3160	**0.2890**	0.3101	0.3184	↓ 0.0270
SupICL	63.90				↑ 0.47	0.2803				↓ 0.0303
+ *Var-IC*	63.90	63.87	63.73	63.83	↓ 0.03	0.2803	0.2293	0.2740	0.2839	↓ 0.0510
+ *Var-Prompt*	63.97	64.33	**64.37**	**64.37**	↑ 0.40	0.3022	0.2736	0.2917	0.3004	↓ 0.0286
+ *Var-Both*	63.97	63.90	64.33	**64.37**	↑ 0.40	0.3022	**0.2500**	0.2853	0.3009	↓ 0.0522

Table 8: Full results of self-ensembling with different variations. We mark the cells of baseline systems without self-ensembling and their results in grey. Numbers in **bold** represents the best metric values for each learning method. Δ calculates the difference of performance and calibration error between the original results (Ori.) and the best self-ensembled results, where green means better results and red means worse results. We run all the experiments above 3 times if possible and show the mean values.

Evaluation Metrics		SST2				SST5			
		ZSL	ICL	SFT	SICL	ZSL	ICL	SFT	SICL
Performance	**acc**	96.38	$96.81_{0.12}$	$96.81_{0.04}$	$96.89_{0.05}$	52.76	$50.59_{0.13}$	$52.93_{0.28}$	$53.47_{0.87}$
	macro f1	-	-	-	-	38.01	$31.28_{0.32}$	$44.65_{0.58}$	$43.91_{1.04}$
+ calibrated	***acc***	96.87	$97.00_{0.03}$	$96.78_{0.05}$	$96.92_{0.08}$	50.59	$52.01_{0.15}$	$51.83_{0.26}$	$51.95_{0.04}$
	macro f1	-	-	-	-	49.33	$50.85_{0.18}$	$51.00_{0.38}$	$51.07_{0.08}$
Trustworthiness	**ECE**	0.9291	$0.9369_{0.0001}$	$0.9340_{0.0011}$	$0.9380_{0.0003}$	0.2309	$0.3065_{0.0018}$	$0.1689_{0.0093}$	$0.1744_{0.0076}$
+ calibrated	***ECE***	0.7937	$0.8000_{0.0002}$	$0.7930_{0.0001}$	$0.7988_{0.0008}$	0.0925	$0.1117_{0.0014}$	$0.1070_{0.0049}$	$0.1067_{0.0014}$
Evaluation Metrics		**Manifestos**				**Hate Speech**			
		ZSL	**ICL**	**SFT**	**SICL**	**ZSL**	**ICL**	**SFT**	**SICL**
Performance	**micro f1**	28.37	$31.37_{0.18}$	$37.04_{0.29}$	$38.25_{0.44}$	54.33	$52.84_{0.06}$	$64.44_{0.51}$	$62.78_{0.58}$
	macro f1	21.71	$25.35_{0.22}$	$33.71_{0.62}$	$35.55_{0.41}$	47.15	$46.49_{0.54}$	$61.72_{0.40}$	$60.27_{0.31}$
+ calibrated	***micro f1***	37.38	$37.96_{0.36}$	$38.75_{0.35}$	$38.83_{0.85}$	46.60	$47.49_{0.33}$	$61.07_{2.57}$	$56.36_{0.64}$
	macro f1	35.26	$35.99_{0.35}$	$36.17_{0.37}$	$36.96_{0.59}$	43.65	$44.78_{0.35}$	$59.23_{2.38}$	$54.80_{0.66}$
Trustworthiness	**ECE**	0.4514	$0.3839_{0.0027}$	$0.1116_{0.0258}$	$0.1066_{0.0153}$	0.2309	$0.1935_{0.0015}$	$0.1047_{0.0446}$	$0.0713_{0.0063}$
+ calibrated	***ECE***	0.0402	$0.0382_{0.0067}$	$0.0447_{0.0063}$	$0.0335_{0.0070}$	0.0956	$0.1076_{0.0017}$	$0.0297_{0.0127}$	$0.0474_{0.0040}$

Table 9: Experiment results with different learning methods using FlanT5-xl. We report the mean value for 3 runs with different random seeds and list the variance in the subscripts. We color the Batch Calibration results in grey.

Systems	SST5									
	Macro F1					ECE				
	Ori.	**Max**	**Mean**	**Majority**	**Δ**	**Ori.**	**Max**	**Mean**	**Majority**	**Δ**
ZSL	38.01				↑ 5.53	0.2309				↓ 0.0898
+ *Var-Prompt*	38.01	**43.54**	41.80	42.65	↑ 5.53	0.2309	**0.1411**	0.1971	0.2210	↓ 0.0898
ICL	31.28				↑ 3.95	0.3065				↓ 0.1171
+ *Var-IC*	31.28	30.57	30.85	30.91	↓ 0.37	0.3065	0.2653	0.3035	0.3070	↓ 0.0412
+ *Var-Prompt*	31.28	34.75	34.71	34.77	↑ 3.49	0.3065	0.2358	0.2714	0.2798	↓ 0.0707
+ *Var-Both*	31.28	**35.23**	33.75	33.51	↑ 3.95	0.3065	**0.1894**	0.2730	0.2846	↓ 0.1171
FT	44.65				↑ 3.33	0.1689				↓ 0.0730
+ *Var-Prompt*	47.06	**47.98**	47.15	47.01	↑ 0.92	0.1656	**0.0959**	0.1425	0.1660	↓ 0.0697
SupICL	43.91				↑ 1.26	0.1744				↓ 0.0740
+ *Var-IC*	43.91	43.91	44.05	44.00	↑ 0.14	0.1744	0.1376	0.1738	0.1742	↓ 0.0368
+ *Var-Prompt*	44.21	44.79	44.69	44.78	↑ 0.58	0.1688	0.1278	0.1536	0.1666	↓ 0.0410
+ *Var-Both*	44.21	**45.17**	44.60	44.41	↑ 0.96	0.1688	**0.1004**	0.1488	0.1638	↓ 0.0684
Systems	**Hate Speech**									
	Macro F1					**ECE**				
	Ori.	**Max**	**Mean**	**Majority**	**Δ**	**Ori.**	**Max**	**Mean**	**Majority**	**Δ**
ZSL	47.15				↑ 0.59	0.2309				↓ 0.0508
+ *Var-Prompt*	47.15	46.57	47.30	**47.74**	↑ 0.59	0.2309	**0.1801**	0.2041	0.2194	↓ 0.0508
ICL	46.49				↑ 1.09	0.1935				↓ 0.1052
+ *Var-IC*	46.49	47.03	46.94	46.74	↑ 0.54	0.1935	0.1301	0.1800	0.1901	↓ 0.0634
+ *Var-Prompt*	46.49	47.44	47.06	47.57	↑ 1.08	0.1935	0.1257	0.1625	0.1836	↓ 0.0678
+ *Var-Both*	46.49	**47.45**	46.80	46.68	↑ 1.09	0.1935	**0.0883**	0.1581	0.1867	↓ 0.1052
FT	61.72				↓ 0.72	0.1074				↓ 0.0572
+ *Var-Prompt*	60.04	60.71	**61.00**	60.86	↑ 0.96	0.0873	**0.0502**	0.0719	0.0887	↓ 0.0371
SupICL	60.27				↑ 1.84	0.0713				↓ 0.0437
+ *Var-IC*	60.27	60.88	60.72	60.68	↑ 0.61	0.0713	0.0424	0.0636	0.0729	↓ 0.0289
+ *Var-Prompt*	60.37	60.97	61.50	61.45	↑ 1.13	0.0883	0.0427	0.0649	0.0793	↓ 0.0456
+ *Var-Both*	60.37	61.69	**62.11**	61.74	↑ 1.74	0.0833	**0.0276**	0.0588	0.0816	↓ 0.0557

Table 10: Results of self-ensembling with different variations using FlanT5-xl. For simplicity, we omit the variance of 3 runs and only show the mean values. The notations follow previous patterns.

D PROMPTING TEMPLATES

D.1 PROMPTING TEMPLATES FOR MAIN EXPERIMENTS

We provide the prompting templates for different datasets when comparing learning methods as follows.

SST-2

```
Classify this sentence’s sentiment into ’positive’ or ’negative’: <
SENTENCE>
<LABEL>
```

RTE

```
Does Sentence1 entails Sentence2?
Sentence1: <SENTENCE1>
SENTENCE2: <SENTENCE2>
<LABEL>
```

ANLI

```
Does the premise entails the hypothesis?
Premise: <PREMISE>
Hypothesis: <HYPOTHESIS>
<LABEL>
```

SST-5

```
Classify this sentence’s sentiment into "terrible", "bad", "neutral",
"good" or "great": <SENTENCE>
<LABEL>
```

NLU++

```
Here is a sentence: ’<SENTENCE>’
Try to answer this question if possible with ’yes’ or ’no’: ’<QUESTION>’
<LABEL>
```

Manifestos

```
Which category about US society does the sentence belong to from "other",
"external relations", "freedom and democracy", "political system",
"economy", "welfare and quality of life", "fabric of society", "social
groups": <SENTENCE>
<LABEL>
```

Hate Speech

```
Classify this sentence’s sentiment into "hate", "neutral" or "support": <
SENTENCE>
<LABEL>
```

D.2 PROMPTING TEMPLATES FOR *Var-Prompt*

Below show the various templates for prompt cycling. All the prompting templates are manually written without specific crafting.

ANLI

```
Does the premise entails the hypothesis?
Premise: <PREMISE>
Hypothesis: <HYPOTHESIS>
<LABEL>
```

```
Premise: <PREMISE>
Hypothesis: <HYPOTHESIS>
Given the premise, is the hypothesis entailed?
<LABEL>
```

```
Is the hypothesis entailed by the premise?
Premise: <PREMISE>
Hypothesis: <HYPOTHESIS>
<LABEL>
```

```
### Instruction: Determine whether the hypothesis is entailed by the
premise.
Premise: <PREMISE>
Hypothesis: <HYPOTHESIS>
<LABEL>
```

SST-5

```
Classify this sentence’s sentiment into "terrible", "bad", "neutral",
"good" or "great": <SENTENCE>
<LABEL>
```

```
<SENTENCE>
is this sentence ’great’, ’good’, ’neutral’, ’bad’ or ’terrible’?
<LABEL>
```

```
<SENTENCE>
Among "terrible", "bad", "neutral", "good" or "great", the sentence’s
sentiment is
<LABEL>
```

```
### Instruction: Classify the input sentence’s sentiment into into
"terrible", "bad", "neutral", "good" or "great".
Input: <SENTENCE>
### Response: <LABEL>
```

Manifestos

```
Which category about US society does the sentence belong to from
"other", "external relations", "freedom and democracy", "political
system", "economy", "welfare and quality of life", "fabric of society",
"social groups": <SENTENCE>
<LABEL>
```

```
<SENTENCE>
Which category about US society does the sentence belong to?
<LABEL>
```

```
<SENTENCE>
Among "other", "external", "democracy", "political", "economy",
"welfare", "fabric", "group", the sentence’s US societal category is
<LABEL>
```

```
### Instruction: Classify the input sentence’s US societal category
into "other", "external", "democracy", "political", "economy",
"welfare", "fabric", "group".
Input: <SENTENCE>
### Response: <LABEL>
```

Hate Speech

```
Classify this sentence’s sentiment into "hate", "neutral" or
"support": <SENTENCE>
<LABEL>

<SENTENCE>
Is the sentence hate, neutral or support?
<LABEL>

<SENTENCE>
Among "hate", "neutral" or "support", the sentence’s sentiment
is <LABEL>

### Instruction: What’s the sentiment of input sentence among "hate",
"neutral" or "support"?
Input: <SENTENCE>
### Response: <LABEL>
```

D.3 PROMPTING TEMPLATES FOR *Var-Prompt* IN ABLATION STUDIES

We use ChatGPT to generate paraphrased prompting templates for *Var-Prompt*. The instruction we give to ChatGPT is as follows.

```
Paraphrase the provided templates and keep the keywords in <> in the
meantime. Show me 5 different paraphrased results.

The template is:
<TEMPLATE>
```

We paraphrase the prompting templates for SST-5 and Hate Speech datasets and randomly sampled 4 paraphrased candidates. These templates share similar wording and structure with the human-written templates. We conduct experiments with these 8 templates in total and use max probability when self-ensembling. We provide the candidates below for reference.

SST-5

```
<SENTENCE>\nThe sentiment expressed by <SENTENCE> falls into the
categories of "terrible," "bad," "neutral," "good," or "great,"
and it is labeled as <LABEL>

Evaluate the sentiment expressed by <SENTENCE>, placing it in
the categories of "terrible," "bad," "neutral," "good," or
"great," and indicate the sentiment as <LABEL>

Evaluate the emotional tone of this statement and categorize it
as "terrible," "bad," "neutral," "good," or "great": <SENTENCE>
<LABEL>

Analyze the emotional inclination of the following statement,
categorizing it as "terrible," "bad," "neutral," "good," or
"great": <SENTENCE>
<LABEL>
```

Hate Speech

```
Assess whether the sentiment in this sentence falls under
"hate," "neutral," or "support": <SENTENCE>
<LABEL>
```

```
Appraise the sentiment expressed in this sentence and assign it
to one of the categories: "hate," "neutral," or "support"
: <SENTENCE>
<LABEL>
```

```
Categorize the sentiment of <SENTENCE> as either "hate," "neutral,"
or "support," with the assigned label being <LABEL>.
```

```
Determine the emotional tone of <SENTENCE>, categorizing it as
"hate," "neutral," or "support," and mark the sentiment as <LABEL>.
```