Calibrating Black Box LLMs with Prompt Variations

Yijin Hua[1], **Justin Svegliato**[2], **Sam Toyer**[2], **Stuart Russell**[2], **Anoop Sinha**[3]
[1]UCLA [2]UC Berkeley [3]Google
yijinhua@g.ucla.edu, {jsvegliato,sdt,russell}@berkeley.edu, anoopsinha@google.com

Abstract

LLMs have shown remarkable success in various natural language tasks. However, because they are still susceptible to factual inaccuracies and hallucinations, it is important to develop techniques that can calibrate these models and measure their confidence, especially in the context of RLHF worsening model calibration. Building on existing work in self-consistency, we introduce a novel technique for calibrating black box LLMs. In particular, given a prompt, our technique can calibrate the LLM by averaging over the token probability distributions generated for a prompt ensemble that is comprised of *prompt variations*. In our preliminary experiments, we show that our technique in combination with temperature scaling considerably decreases expected calibration error and Brier score with minimal impact on accuracy, when using OpenAI's LLM with RLHF `text-davinci-003` to answer questions across a range of QA datasets. The result is an effective way to calibrate black box LLMs without needing to finetune or access model parameters.

Introduction

Large language models (LLMs) have shown remarkable success in various natural language tasks, including dialogue, code completion, language translation, and story generation (Rae et al. 2021; Thoppilan et al. 2022; Ouyang et al. 2022). Nevertheless, when faced with uncommon facts (Poerner, Waltinger, and Schütze 2019) or complex reasoning (Talmor et al. 2020), these models are still susceptible to incorrect answers and hallucinations, generating both minor factual errors and completely fabricated narratives, scenarios, or events. As a result, in order to mitigate the safety risks and consequences of LLMs and responsibly deploy them in real-world applications, it has become increasingly necessary to develop techniques that can be used to calibrate these models and measure their confidence.

In response, there has been work on calibrating LLMs. Generally, these techniques fall into some combination of three approaches. First, in *finetuning* techniques, the model is finetuned using a proxy calibration objective function to directly optimize for model calibration (Jiang et al. 2021; Xiao et al. 2022; Lin, Hilton, and Evans 2022). Next, in *verbal elicitation* techniques, the model provides a verbal confidence by using different prompting strategies that can improve their reasoning and evaluation capabilities (Tian et al. 2023; Xiong et al. 2023). Finally, in *self-consistency* techniques, the model generates a set of answers to different versions of a given prompt to estimate the confidence based on consistency across the set of answers (Zhao et al. 2021; Arora et al. 2023; Xiong et al. 2023; Jiang et al. 2023; Portillo Wightman, Delucia, and Dredze 2023).

Building on existing work in self-consistency, we introduce a novel technique for calibrating black box LLMs that uses a prompt ensemble instead of just a single prompt. To illustrate our technique, consider a question-answering (QA) task in which the LLM must answer a multiple-choice question. To do this, an LLM typically takes in a prompt that includes the multiple-choice question and generates a token probability distribution over each multiple-choice answer, which is often treated as an approximation of confidence. In contrast, our technique (1) generates a prompt ensemble, specifically a set of *prompt variations*, that has identical semantic meaning to the given prompt, (2) feeds the prompt ensemble into the LLM to generate a set of token probability distributions, and (3) averages each token probability distribution to produce a final token probability distribution.

In our preliminary experiments, we use OpenAI's LLM `text-davinci-003`, a model trained with RLHF for instruction following, to answer questions across a range of QA datasets. The key takeaway is that our technique in combination with temperature scaling considerably decreases expected calibration error and Brier score with minimal impact on accuracy. In addition, we observe that the token probability distribution produced by the model spreads out across a range of confidence intervals and the model's calibration approaches perfect model calibration. The result is a novel technique for calibrating black box LLMs that not only improves model calibration but also does not require finetuning the model or accessing the model's parameters and therefore can be easily applied to black box models that are only exposed through a limited API.

Related Work

Calibration of deep learning models has been an area of extensive research (Guo et al. 2017; Wang 2023). For LLMs in particular, (Guo et al. 2017; Desai and Durrett 2020; Kadavath et al. 2022) show that a pretrained LLM's calibration can improve with the number of parameters in the model and that they can be well calibrated after post-hoc calibration methods like temperature scaling. In contrast, LLMs trained with RLHF have been shown to be poorly calibrated even with temperature scaling: (OpenAI 2023) demonstrates that RLHF negatively affects calibration, forcing model logits to concentrate at extreme values and making it difficult to improve model calibration through scaling model logits alone.

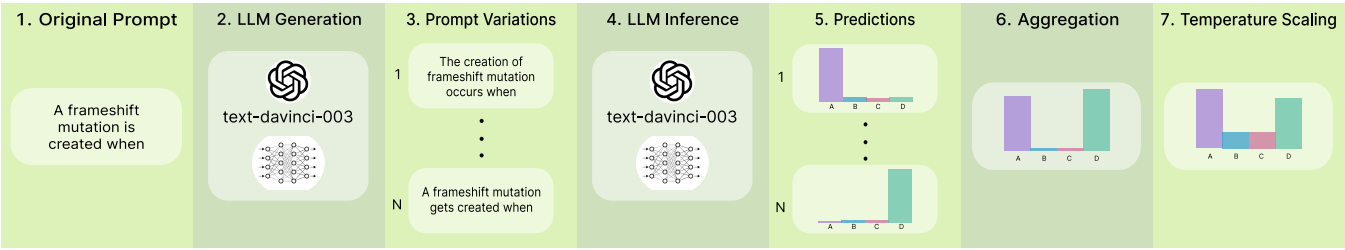


Figure 1: An illustration of our LLM calibration approach.

In general, methods that calibrate LLMs often fall into some combination of each of the three approaches below:

Finetuning In the first approach, the LLM is finetuned using a proxy calibration objective function to directly optimize for model calibration. In general, (Xiao et al. 2022) demonstrates that the choice of proxy objective function can have a significant impact on model calibration. For example, (Jiang et al. 2021) first finetunes the model with a softmax-based or margin-based objective function and then applies posthoc temperature scaling and decision trees. Moreover, (Lin, Hilton, and Evans 2022) first teaches the model to express verbal confidences and then performs supervised finetuning using a mathematical reasoning dataset. In contrast, while these method heavily rely on finetuning, our technique eliminates the need for finetuning entirely.

Verbal Confidence In the next approach, the LLM provides a verbal confidence by using different prompting strategies that can improve their reasoning and evaluation capabilities. This is motivated by model logits being either inaccessible in black box LLMs or rendered inaccurate due to RLHF. For example, (Tian et al. 2023) asks the LLM to elicit verbal confidences for k guesses to a question and averages them together to produce a calibrated verbal confidence. However, this can be very sensitive to the choice of prompt structure, making it difficult to generalize across different tasks, especially those that involve sequential reasoning. To explore this, (Xiong et al. 2023) asks the model to elicit verbal confidences using different temperatures and prompt strategies, including Chain-of-Thought, Multi-Step, and Top-K reasoning. However, while these methods resort to verbal confidences, our technique continues to leverage the confidences encoded in the model logits instead.

Self-Consistency In the final approach, the LLM generates a set of answers to different versions of a given prompt to estimate the confidence based on consistency across the set of answers. Generally, (Zhao et al. 2021; Arora et al. 2023) demonstrate that LLMs can be sensitive to different prompting strategies, example orderings, and question reformulations. Leveraging this idea, (Xiong et al. 2023) asks the model to elicit a verbal confidence for k guesses to a question but represents the question using different prompt strategies and temperatures. Similarly, (Jiang et al. 2023; Portillo Wightman, Delucia, and Dredze 2023) generate an ensemble of prompts with template paraphrasing or option permutations and uses prompt agreement to generate a calibrated confidence. However, while these methods only slightly change the prompt template, insert misleading hints, or permute answer choices, our technique uses entirely different formulations of the prompts themselves.

Original Prompt

```
A frameshift mutation is created when
```

LLM Generation

```
Generate n different ways to ask the question below:
{original question}
```

Prompt Variations

```
1) The creation of frameshift mutation occurs when
.
.
.
N) A frameshift mutation gets created when
```

Figure 2: An idealized prompt structure used in our technique.

Prompt Variations

In this section, we introduce our technique for calibrating black box LLMs. For simplicity, we consider our technique in the setting of QA tasks. In short, for a given prompt, our technique first uses the LLM to generate a prompt ensemble that consists of a set of *prompt variations*, each of which have identical semantic meaning to the given prompt. After that, for each prompt variation in the prompt ensemble, our technique uses the LLM to compute a token probability distribution. Finally, the token probability distributions for each prompt variation in the prompt ensemble are averaged to produce a final token probability distribution. In our experiments, we show that this leads to final token probability distributions that get close to perfect model calibration.

Intuitively, we hypothesize that calibrating an LLM with prompt variations works for similar reasons to typical ensemble approaches in machine learning. By using prompt variations, the activations of the LLM can potentially travel through different regions/trajectories of the neural network. Naturally, each region/trajectory may assign too much or too little confidence to a specific answer. Consequently, by using the token probability distributions from a range of prompt variations, the overconfidence and underconfidence can be mitigated, resulting in a better calibrated LLM.

Notation We begin by describing some notation that will be used in the paper. Formally, given a vocabulary T, an LLM takes in an input token sequence $t_1, \ldots, t_n$ and outputs a token probability distribution $\Pr(t|t_1, \ldots, t_n)$ over each

Metric	Method	OpenbookQA	TruthfulQA	LogiQA	MMLU-Bio	MMLU-Nutrition	MMLU-History	MMLU-Physics	MMLU-Law	MMLU-Marketing	All
ECE ↓	∅	0.111	0.235	0.232	0.087	0.117	0.112	0.195	0.097	0.037	0.141
	[VAR]	0.084	0.206	0.205	0.085	0.086	0.077	0.138	0.093	0.047	0.116
	[TEMP]	0.059	0.174	0.151	0.047	0.075	0.079	0.130	0.066	**0.024**	0.090
	[VAR, TEMP]	**0.033**	**0.122**	**0.109**	**0.031**	**0.025**	**0.054**	**0.072**	**0.032**	0.037	**0.039**
BRIER ↓	∅	0.25	0.47	0.50	0.23	0.25	0.29	0.43	0.27	0.13	0.32
	[VAR]	0.22	0.41	0.40	0.20	0.23	0.23	0.33	0.24	0.12	0.27
	[TEMP]	**0.18**	0.31	0.34	**0.17**	**0.19**	0.21	0.30	0.19	**0.09**	0.22
	[VAR, TEMP]	0.19	**0.27**	**0.28**	**0.17**	**0.19**	**0.20**	**0.26**	**0.18**	**0.09**	**0.21**
ACCURACY ↑	∅	**0.71**	**0.47**	**0.45**	**0.77**	**0.72**	**0.69**	**0.51**	**0.70**	**0.85**	**0.64**
	[VAR]	0.69	0.43	**0.45**	0.75	0.65	**0.69**	0.50	**0.70**	0.81	0.62
	[TEMP]	**0.71**	**0.47**	**0.45**	**0.77**	**0.72**	**0.69**	**0.51**	**0.70**	**0.85**	**0.64**
	[VAR, TEMP]	0.69	0.43	**0.45**	0.75	0.65	**0.69**	0.50	**0.70**	0.81	0.62

Table 1: The expected calibration error, Brier score, and accuracy for each LLM calibration method.

output token t. Specifically, given the input token sequence $t_1, \ldots, t_n$, the LLM first generates a score z_t for each output token t and then generates the token probability distribution $\Pr(t|t_1, \ldots, t_n)$ over each output token t by using the softmax probability function in the following way:

$$\Pr(t|t_1, \ldots, t_n) = \frac{e^{z_t}}{\sum_{t'} e^{z_{t'}}}.$$

In the setting of QA tasks in particular, a multiple choice question Q is an input token sequence $Q = \langle t_1, \ldots, t_n \rangle$ and the multiple choice answers A is a set of output tokens $A = \{1, 2, \ldots, \ell\}$. Consequently, given a multiple choice question $Q = \langle t_1, \ldots, t_n \rangle$, the LLM predicts a multiple choice answer $\hat{a}$ in the following way:

$$\hat{a} = \operatorname*{argmax}_a \Pr(a|Q).$$

The goal of this paper is to calibrate the LLM. Here, *calibration* can be viewed as the process of ensuring that the token probability distribution generated by the LLM matches the probability of the LLM being correct. In QA tasks specifically, if the LLM generates a probability p for a multiple choice answer a given a multiple choice question Q, the LLM should be correct approximately p of the time. Generally, this enables the token probability distribution generated by the LLM to be a decent approximation of confidence.

Methodology We now describe our technique for calibrating a black box LLM with a prompt ensemble. In our technique, a prompt ensemble $E(P)$ is a set of prompt variations that have identical semantic meaning to a given prompt P. In the setting of QA tasks, a prompt ensemble $E(Q)$ is therefore a set of multiple choice question variations that have identical semantic meaning to a given multiple choice question Q. As an example, suppose we have the multiple choice question Q = "*What is the capital city of the US?*". Given this, the prompt ensemble $E(Q)$ could be composed of the prompt variations "*Can you tell me the name of the capital city of the US?*", "*Where is the capital city of the US located?*", and "*What is the name of the US capital?*".

In order to generate the prompt ensemble for a given prompt, a range of methods can be used. For our technique in particular, the LLM `text-davinci-003` is used to generate the prompt ensemble as it is a standard black box LLM that is available for public use. By using a specific prompt structure for different formats of multiple choice questions (e.g. *fill-in-the-blank*, *all-that-apply*, *all...except*), the LLM can accurately generate a prompt ensemble $E(Q)$ for a given multiple choice question Q. We provide an idealized prompt template used by our technique for a fill-in-the-blank multiple choice question in Figure 2.

After generating the prompt ensemble $E(Q)$ for a given multiple choice question Q, each multiple choice question variation $e \in E(Q)$ is input into the LLM and a token probability distribution $\Pr_e(a|e)$ is output by the LLM. The token probability distributions $\Pr_e(a|e)$ for each prompt $e \in E(Q)$ is then averaged together to generate a final token probability distribution $\Pr(a|Q)$. In our experiments, we demonstrate that the final token probability distribution $\Pr(a|Q)$ is better calibrated compared to each individual token probability distributions $\Pr_e(a|e)$.

Figure 1 illustrates our technique. In Steps (1) to (3), a prompt ensemble $E(P)$ is generated by using the LLM for a given prompt P. In Steps (4) to (5), the LLM generates a set of token probability distributions $\Pr_e(a|e)$ for each prompt variation e within the prompt ensemble $E(P)$. In Step (6), these predicted token distributions $\Pr_e(a|e)$ are averaged together to generate a final token probability distribution $\Pr(a|Q)$. In Step (7), it is possible to apply temperature scaling to the final token probability distribution $\Pr(a|Q)$.

Temperature Scaling Our technique can be combined with *temperature scaling* (Guo et al. 2017), which is a common, simple, and effective calibration method. In temperature scaling, the token probability distribution generated by the LLM is adjusted by introducing a temperature constant τ, making the token probability distribution more or less uniform (spread out). Formally, the token probability distribution $\Pr(t|t_1, \ldots, t_n)$ is adjusted by introducing a temperature constant τ in the following way:

$$\Pr(t|t_1, \ldots, t_n) = \frac{e^{z_t/\tau}}{\sum_{t'} e^{z_{t'}/\tau}}.$$

There are multiple calibration methods that may be compatible with our technique, such as *Platt scaling* (Böken 2021) and *isotonic regression* (Zadrozny and Elkan 2002), but we consider temperature scaling here due to its simplicity.

Preliminary Experiments

Our experiments use OpenAI's LLM `text-davinci-003`, a model trained for instruction following using RLHF, to answer questions across a range of QA datasets. This LLM

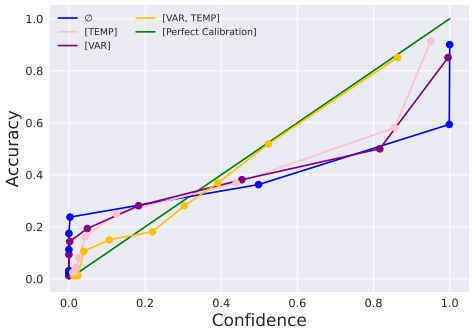


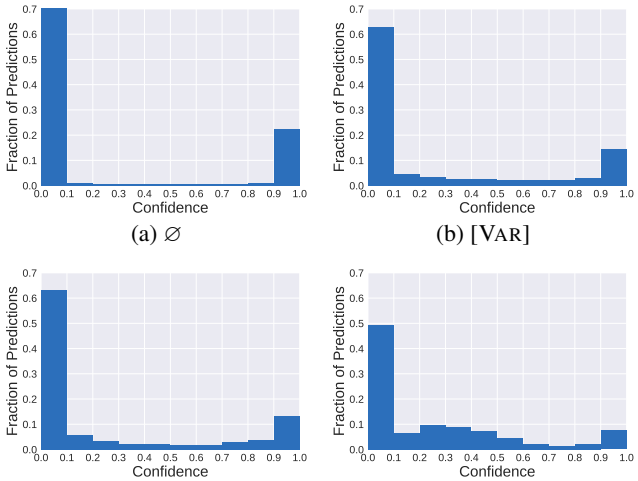
Figure 3: The reliability diagram for each LLM calibration method.

is selected because it is a typical black box LLM that can only be accessed through an API that provides model logits for the top k token completions. Here, we compare the following LLM calibration methods: *no calibration* ($\varnothing$), *prompt variations* ([VAR]), *temperature scaling* ([TEMP]), and *prompt variations with temperature scaling* ([VAR, TEMP]). Overall, our experiments provide encouraging results that *prompt variations with temperature scaling* significantly boost model calibration.

Datasets We evaluate model calibration on a range of QA datasets. **OpenbookQA** (Mihaylov et al. 2018) contains questions that involve combining commonsense knowledge with open book facts. **TruthfulQA** (Lin, Hilton, and Evans 2021) contains questions associated with common misconceptions and false beliefs held by people. **LogiQA** (Liu et al. 2020)) contains questions that involve logical reasoning. **MMLU** (Hendrycks et al. 2021) contains a set of over 50 datasets for different subject areas (specifically including Bio, Nutrition, History, Physics, Law, and Marketing). Note that a collection of 100 questions are randomly sampled from each of the 9 datasets considered in our experiments for a total of 900 questions.

Metrics We evaluate model calibration along several metrics. **Expected Calibration Error** (ECE) is the expected absolute difference—over N confidence bins—between the actual accuracy of a bin and a given model's mean token probability for a bin: $\text{ECE} = \frac{1}{N}\sum_{i=1}^{N} |B_i| \cdot |\text{Accuracy}(B_i) - \text{Confidence}(B_i)|$. Note that the token probabilities for all answers are binned into 10 bins of equal size. **Brier Score** (BRIER) is the expected squared difference between a given model's token probabilities and the actual correctness (0 or 1), which penalizes random guessing from the model. **Accuracy** (ACC) is the number of questions answered correctly divided by the total number of questions.

Results Figure 3 shows the reliability diagram for each LLM calibration method. For each confidence bin, this figure plots the mean predicted probability against the actual accuracy. Therefore, the objective is to achieve perfect model calibration (*green*). Here, we see that the baseline LLM without any calibration (*blue*) is quite miscalibrated with an accuracy ranging between 0.6 and 0.9 when the confidence is near 1.0 and between 0.0 and 0.3 when the confidence is near 0.0. This is due to RLHF forcing model logits to concentrate at the extreme values. Moreover, prompt variations (*purple*) and temperature scaling (*pink*) used in isolation tend to improve model calibration. Finally, when using prompt variations with temperature scaling (yellow), the LLM gets close to perfect model calibration (*green*).



(a) $\varnothing$ (b) [VAR] (c) [TEMP] (d) [VAR, TEMP]

Figure 4: The frequency diagrams over the token probabilities.

Table 1 show the expected calibration error, Brier score, and accuracy metrics for each LLM calibration method. Here, we find that prompt variations with temperature scaling ([VAR, TEMP]) consistently decreases the expected calibration error by over 70% and the Brier score by over 34% across all QA datasets. Moreover, we see that accuracy remains roughly the same despite the decrease in the expected calibration error and the Brier score, which is encouraging.

Figure 4 has the frequency diagrams over the token probabilities generated by each LLM calibration method. First, for $\varnothing$, the token probabilities are heavily concentrated at the extreme values of 0.0 and 1.0 confidence intervals, which is expected as RLHF encourages the model to assign the most weight to the answer with the highest probability instead of matching the model logits with the probability of being correct. Next, for [TEMP] and [VAR], the token probabilities further spread out across different confidence intervals because [TEMP] smooths out token probabilities and [VAR] averages token probabilities across the different prompt variations. Finally, for [VAR, TEMP], the token probabilities are even further spread out across the confidence intervals.

Conclusion

In this paper, we introduce a novel technique for calibrating black box LLMs. In particular, given a prompt, our technique improves LLM calibration by averaging over the token probability distributions generated for a prompt ensemble that is comprised of prompt variations. In our experiments, we demonstrate that our technique in combination with temperature scaling considerably decreases expected calibration error and Brier score with minimal impact on accuracy. Future work will develop different classes of prompt variations and apply our technique to a variety of models and datasets.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work was supported by a gift from the Open Philanthropy Foundation.

References

Arora, S.; Narayan, A.; Chen, M. F.; Orr, L.; Guha, N.; Bhatia, K.; Chami, I.; Sala, F.; and Ré, C. 2023. Ask Me Anything: A simple strategy for prompting language models. *International Conference on Machine Learning*.

Böken, B. 2021. On the appropriateness of Platt scaling in classifier calibration. *Information Systems*.

Desai, S.; and Durrett, G. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

Guo, C.; Pleiss, G.; Sun, Y.; et al. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*.

Hendrycks, D.; Burns, C.; Basart, S.; et al. 2021. Measuring massive multitask language understanding. *International Conference on Learning Representations*.

Jiang, M.; Ruan, Y.; Huang, S.; Liao, S.; Pitis, S.; Grosse, R. B.; and Ba, J. 2023. Calibrating language models via augmented prompt ensembles. *ICML Workshop on Challenges in Deployable Generative AI*.

Jiang, Z.; Araki, J.; Ding, H.; and Neubig, G. 2021. How can we know when language models know? *Association for Computational Linguistics*.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez; et al. 2022. Language models (mostly) know what they know. *Association for Computational Linguistics*.

Lin, S.; Hilton, J.; and Evans, O. 2021. TruthfulQA: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Lin, S.; Hilton, J.; and Evans, O. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Liu, J.; Cui, L.; Liu, H.; et al. 2020. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. *Empirical Methods in Natural Language Processing*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Ouyang, L.; Wu, J.; Jiang, X.; et al. 2022. Training language models to follow instructions with human feedback. *Neural Information Processing Systems*.

Poerner, N.; Waltinger, U.; and Schütze, H. 2019. E-BERT: Efficient-yet-effective entity embeddings for BERT. *arXiv preprint arXiv:1911.03681*.

Portillo Wightman, G.; Delucia, A.; and Dredze, M. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. *Association for Computational Linguistics*.

Rae, J. W.; Borgeaud, S.; Cai, T.; et al. 2021. Scaling language models: Methods, analysis, and insights from training Gopher. *arXiv preprint arXiv:2112.11446*.

Talmor, A.; Elazar, Y.; Goldberg, Y.; and Berant, J. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*.

Thoppilan, R.; De Freitas, D.; Hall, J.; et al. 2022. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Tian, K.; Mitchell, E.; Zhou, A.; Sharma, A.; Rafailov, R.; Yao, H.; Finn, C.; and Manning, C. D. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Wang, C. 2023. Calibration in deep learning: A Survey of the atate-of-the-art. *arXiv preprint arXiv:2308.01222*.

Xiao, Y.; Liang, P. P.; Bhatt, U.; Neiswanger, W.; Salakhutdinov, R.; and Morency, L.-P. 2022. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *Association for Computational Linguistics*.

Xiong, M.; Hu, Z.; Lu, X.; Li, Y.; Fu, J.; He, J.; and Hooi, B. 2023. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. *arXiv preprint arXiv:2306.13063*.

Zadrozny, B.; and Elkan, C. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Zhao, Z.; Wallace, E.; Feng, S.; Klein, D.; and Singh, S. 2021. Calibrate before use: Improving few-shot performance of language models. *International Conference on Machine Learning*.