Calibrating Large Language Models with Sample Consistency

Qing Lyu[*1], **Kumar Shridhar**[*2], **Chaitanya Malaviya**[1], **Li Zhang**[1], **Yanai Elazar**[3], **Niket Tandon**[3], **Marianna Apidianaki**[1], **Mrinmaya Sachan**[2], **Chris Callison-Burch**[1]

[1]University of Pennsylvania
[2]ETH Zurich,
[3]Allen Institute for AI
lyuqing@sas.upenn.edu, shkumar@ethz.ch

Abstract

Accurately gauging the confidence level of Large Language Models' (LLMs) predictions is pivotal for their reliable application. However, LLMs are often inherently uncalibrated and elude conventional calibration techniques due to their proprietary nature and massive scale. In this work, we derive model confidence from the distribution of multiple randomly sampled generations, using three measures of *consistency*. We extensively evaluate eleven open and closed-source models on nine reasoning datasets. Results show that consistency-based calibration methods outperform existing post-hoc approaches in terms of calibration error. Meanwhile, we find that factors such as intermediate explanations, model scaling, and larger sample sizes enhance calibration, while instruction-tuning makes calibration more difficult. Moreover, confidence scores obtained from consistency can potentially enhance model performance. Finally, we offer guidance on choosing suitable consistency metrics for calibration, tailored to model characteristics such as the exposure to instruction-tuning and RLHF.

Code — https://github.com/veronica320/Calibrating-LLMs-with-Consistency

Extended version — https://arxiv.org/abs/2402.13904

1 Introduction

Large Language Models (LLMs) excel in various tasks, yet it is hard to know when they err. A first step towards making LLMs more trustworthy is for them to provide a confidence estimate with predictions (Papadopoulos, Edwards, and Murray 2001). This estimate needs to be *calibrated*, meaning that the confidence level is aligned with the likelihood of the prediction being correct (Brier 1950). A well-calibrated system can enable model developers to provide selective predictions, help users decide when to trust or distrust model responses, and potentially facilitate performance improvement through human intervention or self-refinement (Madaan et al. 2023; Shridhar et al. 2023).

Unfortunately, LLMs are not well-calibrated off-the-shelf — the probability logits of model predictions are often poorly aligned with performance (Jiang et al. 2021; Chen et al. 2023).

∗ Equal contribution. Qing Lyu did her work while interning at Allen Institute for AI.

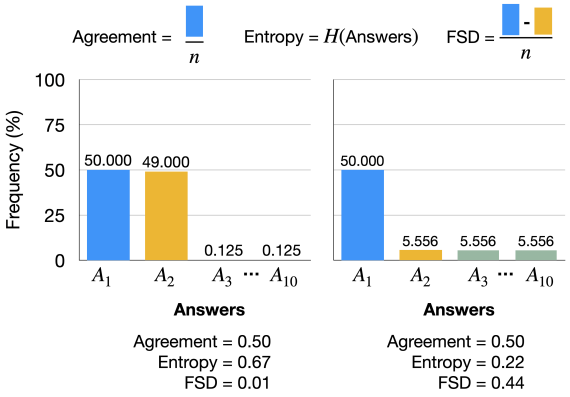

Figure 1: We study three consistency measures, agreement-based, entropy-based, and first-second-distance-based (FSD), to estimate confidence from model generation distributions.

While traditional calibration methods (Guo et al. 2017; Lakshminarayanan, Pritzel, and Blundell 2017; Gal and Ghahramani 2016, i.a.) can be used on open-source LMs, for recent LLMs, these methods become formidably costly because of the need to retrain multiple copies of the model, and might even be inapplicable due to inaccessible training data, model weights, and output probabilities in closed-source LLMs.

In light of these issues, a recent line of work measures the *consistency* of model generations to calibrate confidence (Wang et al. 2023; Xiong et al. 2023, i.a.), with the advantage of being fully post-hoc and requiring no additional calibration data. However, existing work has only used the *agreement* between the original generation and multiple randomly sampled generations as a metric for consistency, ignoring the rich information from the *distribution* of generations.

In this work, we investigate the research question: *How can we best elicit a model's confidence from the consistency of multiple generations?* As shown in Figure 1, we consider three ways to measure consistency, focusing on different characteristics of the distribution: **agreement-based**, as mentioned before; **entropy-based**, which is based on the normalized entropy of the generation distribution; and **FSD-based**, which measures the percentage difference in samples agreeing with the majority and second-majority answers. For exam-

ple, consider two distributions over ten possible answer options (A_1 to A_{10}) in Figure 1. In the left distribution, A_1 and A_2 are almost equally frequent (50% vs. 49%), with the remaining 1% mass equally divided among the rest. In the right distribution, A_1 is still the most frequent (50%), while A_2 through A_{10} are equally frequent. Agreement-based consistency would provide the same confidence estimate (0.50) for both distributions, whereas FSD and entropy can distinguish between them by not relying only on the most popular answer.

We study the effectiveness of each consistency metric when applied to confidence calibration on both open-source (LLAMA, Mistral, Olmo) and closed-source LLMs (Codex, GPT-3.5-turbo, GPT-4), and on nine datasets of four diverse reasoning tasks (Math Reasoning, Multi-Hop QA, Planning, Relational Reasoning). Our experiments reveal several interesting findings: (i) On average, all three consistency metrics significantly outperform existing post-hoc calibration baselines such as probabilistic and verbalized confidence extraction methods (Kadavath et al. 2022; Lin, Hilton, and Evans 2022). (ii) When prompted to generate explanations before the answer, large models exhibit markedly improved calibration. (iii) Scaling model size enhances calibration, whereas instruction-tuning shows a negative effect. Increasing the number of generation samples leads to more accurate calibration, though notable improvements can be observed with as few as 3-5 samples. (iv) We show in an oracle case study that consistency not only offers more reliable confidence estimates, but also holds the potential to enhance model performance on end tasks.

Our contributions are as follows: First, we systematically study three approaches for confidence calibration through sample consistency, and validate their superiority compared to existing post-hoc calibration baselines. Second, we provide a detailed analysis of factors influencing calibration properties of LLMs, especially the role of prompting strategies. Third, we provide researchers with a flow chart to help them pick the most effective consistency measure based on the characteristics of their model.

2 Related Work

Confidence Calibration in LMs. Traditional calibration methods, such as probabilistic (Guo et al. 2017), ensemble-based (Lakshminarayanan, Pritzel, and Blundell 2017; Gal and Ghahramani 2016), and density-based approaches (Lee et al. 2018; Yoo et al. 2022), have proved effective in better calibrating the confidence in white-box LMs. These methods require access to the model logits and/or their pretraining data, involve retraining multiple copies of the same model, or necessitate another dedicated calibration dataset. With the advent of LLMs, they become overly expensive and sometimes even inapplicable to closed-source LLMs. To this end, several post-hoc approaches have been developed. Kadavath et al. (2022) prompt the model to estimate the probability of its generated response being "True", while Lin, Hilton, and Evans (2022) and Mielke et al. (2022) investigate whether the model can directly verbalize its confidence (e.g., "highly confident", or "80% confident"). Another line of work focuses on calibrating confidence with sample consistency (Wang et al. 2023; Manakul, Liusie, and Gales 2023; Xiong et al. 2023; Portillo Wightman, Delucia, and Dredze 2023, i.a.), which only needs input and output access to the model. However, existing studies have only focused on agreement-based measures of consistency, which cannot distinguish between certain distributions (e.g. in Figure 1) since it only relies on the most popular answer. This necessitates a systematic study on how to best elicit confidence from consistency.

Consistency. The term "consistency" has been used to refer to multiple concepts in NLP, including factual alignment (Tam et al. 2022), logical soundness (Nye et al. 2021), agreement within diverse outputs (Wang et al. 2023), among others. We use the term "consistency" to refer to the uniformity in the distribution of multiple model generations, as measured by three metrics in Figure 1.

Reasoning Strategies in LLMs. LLMs exhibit impressive reasoning capabilities with in-context learning. Besides standard prompting (Brown et al. 2020), explanation-based prompting, where models produce a reasoning chain before the answer, brings a notable performance gain. The explanation can be in the form of free-text (Wei et al. 2022), decomposed subquestions (Shridhar et al. 2022; Zhou et al. 2023), or structured symbolic language (Chen et al. 2022; Lyu et al. 2023). We study how calibration can be influenced by representative strategies from each category.

3 Method

Consistency over multiple generations can be used as an indicator for understanding the confidence associated with model predictions. It has been studied in the past for logit-based uncertainty estimation such as model ensembling (Lakshminarayanan, Pritzel, and Blundell 2017) and we extend it to multiple generations in LLMs. For a given input x, we sample a set of n candidate outputs $\hat{s}_1, \dots \hat{s}_n$ using a temperature $T > 0$. From each output $\hat{s}_i$, we extract the answer $\hat{a_i}$ using task-dependent regular expressions. We do a majority voting over the entire answer (multi-)set $\mathbf{a} = \{\hat{a_1} \dots \hat{a_n}\}$ to get the most-voted answer $\bar{a} = \arg\max_a \sum_{i=1}^{n} \mathbb{1}(\hat{a_i} = a)$, where a takes on values from the set of unique answers $\bar{\mathbf{a}}$.

We discuss three ways to measure consistency: agreement-based, entropy-based, and FSD-based. From each measure, we will obtain a confidence score $\texttt{conf}(x, \bar{a})$ for each input x to calibrate the correctness of the prediction.

Agreement-based. Following previous work (Wang et al. 2023; Xiong et al. 2023), we compute the agreement-based consistency by calculating the percentage of answers in $\mathbf{a}$ that agree with the most-voted answer $\bar{a}$. In other words, agreement-based consistency, $\texttt{Agree}(\bar{a})$ is defined as:

$$\texttt{Agree}(\bar{a}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(\hat{a_i} = \bar{a}) \tag{1}$$

Entropy-based. In classification tasks, the entropy of output class probabilities has been used to estimate prediction uncertainty (Gal 2016). We extend this idea to the distribution of multiple model generations to understand the uncertainty in solving an open-ended reasoning problem, where a lower entropy indicates a more consistent distribution.

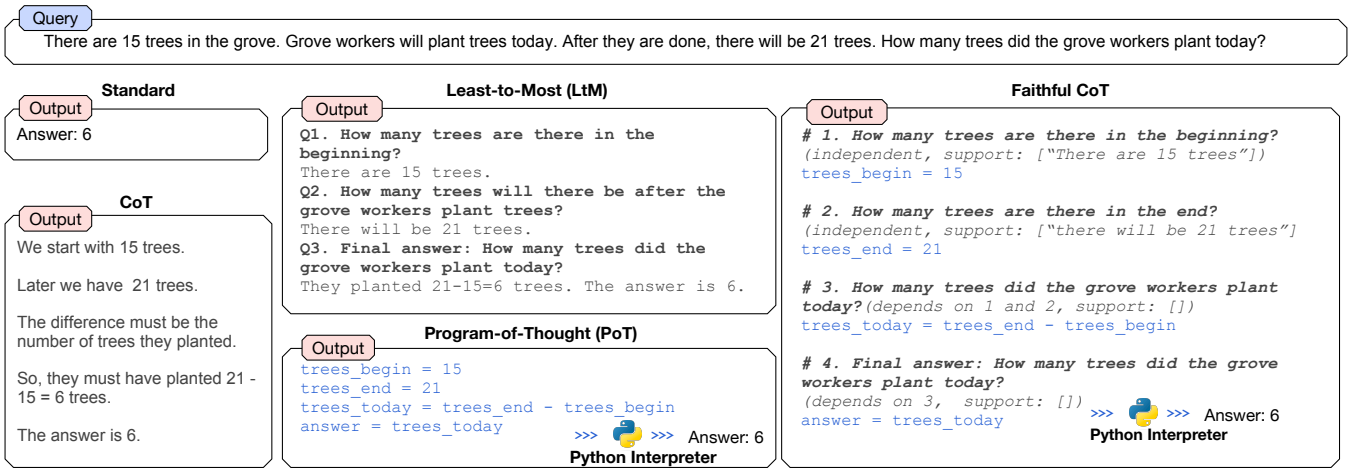

Figure 2: We study how prompting strategies (standard vs. four explanation-based) affect confidence calibration. We provide an example of a math question and showcase the outputs by the five prompting strategies we consider.

To calculate entropy-based consistency, we first obtain a set of answers without duplicates $\bar{\mathbf{a}}$. Then, we define entropy-based consistency, $\texttt{Ent}(\mathbf{a})$ as:

$$\texttt{Ent}(\mathbf{a}) = 1 - \left(-\frac{1}{\log(|\bar{\mathbf{a}}|)} \sum_{i=1}^{|\bar{\mathbf{a}}|} p_i \log(p_i)\right) \quad (2)$$

where, the cardinality of the unique answer set $|\bar{\mathbf{a}}|$ denotes the number of unique answers in the set $\mathbf{a}$ and the probability p_i is the normalized frequency of each unique answer $\bar{a}_i$ in the multi-set $\mathbf{a}$.

Note that the normalized entropy on the right side of the equation is subtracted from 1 to reverse the range between $[0, 1]$ as the lower the entropy, the more consistent the samples are, and thereby the higher the elicited confidence is.

FSD-based. Since the entropy-based measure considers all unique answers that might be skewed toward the tail of the frequency distribution, and agreement-based consistency relies on the most-voted answer, we propose a third alternative, FSD. To compute FSD-based consistency, we consider the top two most-voted answers ($\bar{a}$ and $\bar{\bar{a}}$) and calculate the corresponding agreements $\texttt{Agree}(\bar{a})$ and $\texttt{Agree}(\bar{\bar{a}})$. Then, we use the difference between the two to compute the FSD-based consistency, $\texttt{FSD}(\mathbf{a})$:

$$\texttt{FSD}(\mathbf{a}) = \texttt{Agree}(\bar{a}) - \texttt{Agree}(\bar{\bar{a}}) \quad (3)$$

This metric is particularly useful for cases when the model is unsure about the most-voted answer and places high confidence in the top two predictions (Figure 1 left). In such cases, an FSD-based consistency measure can avoid overconfidence based on the most-voted answer alone.

4 Experimental Setup

Baselines. We compare consistency-based calibration with four post-hoc methods:[1]

• **Raw logits** (**logit**) directly considers the probability of the generation as the confidence. Specifically, we take the exponential of the average log probability of all tokens in the output sequence, which is equivalent to the reciprocal of perplexity.

• **P(True)** (Kadavath et al. 2022) prompts the model to judge the truthfulness of its generation and considers the normalized probability assigned to the 'True' token as its confidence. Our experiments consider both 0-shot and 8-shot prompting (**ptrue**$_{\text{0-shot}}$ and **ptrue**$_{\text{8-shot}}$).

• **Verbalized Confidence** (Lin, Hilton, and Evans 2022) prompts the model to explicitly verbalize its confidence in its generation as a linguistic expression (**verb**$_{\text{ling}}$) from "almost no chance", "likely", ..., to "almost certain", which are mapped to a confidence level; or a percentage (**verb**$_{\text{percent}}$) from 0 to 100, directly used as the confidence score.

We compare consistency-based calibration with only verbalized methods for GPT-3.5-turbo and GPT-4 since probabilities for top-k generations are not accessible, and with only logit for open-source models due to high computation cost (see details in Appendix 2.2).

Tasks. We experiment with 9 datasets from 4 reasoning tasks following previous work (Wei et al. 2022; Lyu et al. 2023):[2]

• **Math Word Problems (MWPs)**: ASDiv (Miao, Liang, and Su 2020), GSM8K (Cobbe et al. 2021), MultiArith (Roy and Roth 2015), and SVAMP (Patel, Bhattamishra, and Goyal 2021).

• **Multi-hop QA**: StrategyQA (Geva et al. 2021), and two BIG-BENCH datasets (Srivastava et al. 2022), Date Understanding and Sports Understanding.

• **Planning**: SayCan (Brohan et al. 2023).

• **Relational inference**: CLUTRR (Sinha et al. 2019).

[1] See Appendix 3 for detailed descriptions and sample prompts. Due to space limit, the appendix can be found in the extended version linked in the Abstract.

[2] See Appendix 4 for dataset statistics and examples.

LM	Consistency Metrics			Baselines				
	entropy	agreement	FSD	$\text{verb}_{\text{ling}}$	$\text{verb}_{\text{percent}}$	logit	$\text{ptrue}_{\text{0-shot}}$	$\text{ptrue}_{\text{8-shot}}$
Codex	.175	**.151**†	.159†	.249	.249	.209	.188	.179
GPT-3.5-turbo	**.205**†	.221†	.207†	.271	.273	n/a	n/a	n/a
GPT-4	.116†	.119†	**.114**†	.154	.181	n/a	n/a	n/a

Table 1: Consistency metrics result in better Brier Scores than baselines (↓) for closed-source models. Scores are averaged across four domains and five prompting strategies. The best scores are **in bold** and the second-best scores are underlined. † indicates that the consistency metric performs statistically significantly better than the best baseline ($p < 0.05$ under paired t-test).

LM	Consistency Metrics			Baselines
	entropy	agree	FSD	logit
LLaMA-7B	.241†	**.232**†	.235†	.474
LLaMA-13B	.222†	**.204**†	.211†	.389
LLaMA-70B	.182†	**.154**†	.165†	.252
Mistral-7B	.205†	**.183**†	.191†	.324
Mistral-7B-it	.220†	.216†	**.215**†	.384
Olmo-7B	.240†	**.202**†	.223†	.514
Olmo-7B-it	.250†	**.239**†	.246†	.478
Olmo-7B-it-rl	**.253**†	.268†	.259†	.523

Table 2: Consistency metrics result in better Brier Scores (↓) than the logit baseline for open-source models.

Evaluation metrics. We use two established calibration error metrics following (Geng et al. 2023), **Brier Score** (BS) (Brier 1950) and **Expected Calibration Error (ECE)** (Guo et al. 2017). Let $\mathcal{D} = \{(x_j, y_j)\}, j \in \{1, \ldots, N\}$ be the evaluation set used to measure calibration. Here x_j's are inputs and y_j's are ground-truth answers. Brier Score measures the mean squared error between the confidence and the prediction correctness:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(\texttt{conf}(x_j, \hat{y}_j) - \mathbb{I}(\hat{y}_j = y_j))^2 \quad (4)$$

where the indicator $\mathbb{I}(\cdot)$ equals 1 when the prediction is correct, and otherwise it is 0.

Since ECE has known issues such as sensitivity to the bin size (Geng et al. 2023), we use Brier Score as the main metric and leave the ECE definition and results in the Appendix.

Prompting strategies. We compare five prompting strategies showcased in Figure 2: **standard** prompting, where an exemplar contains only the query and the answer; **CoT** (Wei et al. 2022), which additionally includes a Natural Language (NL) reasoning chain; **Least-to-Most (LtM)** (Zhou et al. 2023), which decomposes the question into NL subquestions; **Program of Thoughts (PoT)**[3] (Chen et al. 2023), which solves the query in Symbolic Language (SL); and **Faithful CoT (FCoT)** (Lyu et al. 2023), which interleaves NL subquestions and SL solutions. We use the same prompts from Lyu et al. (2023), with the same number of shots for each strategy (6 to 10, depending on the dataset), with only exception being Olmo models where we used 4-shot prompts due to their context length restriction to 2K tokens.

[3]Also called Program-Aided Language Model (PAL) in the concurrent work by Gao et al. (2023).

LMs. We consider the following LLMs: LLaMA (7B/13B/70B), Mistral (7B/7B-it), Olmo(7B/7B-it/7B-it-rl) Codex, GPT-3.5-turbo, and GPT-4. Specifically, "it" stands for instruction-tuning, and "rl" stands for Reinforcement learning from Human Feedback (RLHF).[4]

Sampling Strategy. We sample $n = 40$ candidate outputs with a temperature of $T = 0.4$ for each input following Lyu et al. (2023) in Section 5, and analyze other values of n in Section 6. We select the majority-voted answer as the final answer, following Wang et al. (2023).

5 Results

We study our research question – *how can we best elicit a model's confidence from the consistency of multiple generations?* – from two perspectives: which **calibration method** is the most effective, and how does the **prompting strategy** affect a model's calibration properties?

5.1 Comparing Calibration Methods

We compare all calibration methods in Table 1 and 2, which show the Brier Score for closed-source and open-source LMs averaged across datasets. See full results in Appendix 5.3.

Consistency-based methods are more effective than baselines. Our results suggest a clear advantage of consistency-based calibration methods over the baselines. Averaging across domains, all three consistency metrics almost always result in a significantly lower Brier Score ($p < 0.05$) than the best-performing baseline. This trend also holds across the vast majority of the LMs and domains tested. In rare exceptions in the Relational Inference and Planning domains, the optimal consistency metric often performs statistically the same as the baseline.

Agreement-based consistency works best for open-source models and Codex, while FSD and entropy for the other closed-source models. Among all three consistency metrics, which one is the most effective? We compare the statistical significance between the performance differences of the three metrics in Table 3 in Appendix 5. For closed-source models, agreement is the best metric for Codex ($p < 0.05$), while entropy and FSD are closely competing within a negligible performance gap ($\delta_{BS} \leq 0.002$, $p \geq 0.05$) for GPT-3.5-turbo and GPT-4. Meanwhile, open-source models predominantly favor agreement ($p < 0.05$), with FSD closely following as the second-best metric. The sole exception to

[4]See checkpoints and computational resources in Appendix 2.

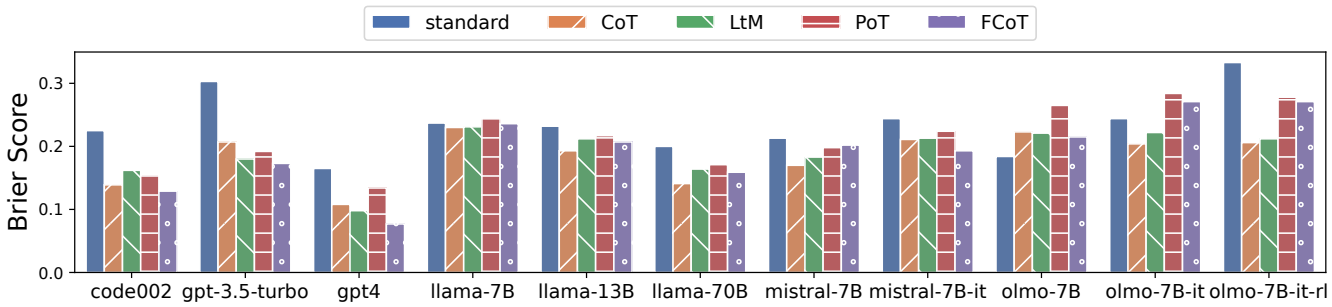

Figure 3: Brier Scores (↓) are improved with explanation-based prompting strategies, especially for larger models. Scores here are averaged across all datasets and consistency metrics.

this trend is in the case of Mistral-7B-it, where FSD leads over agreement by a slim margin (0.215 vs. 0.216, $p \geq 0.05$).

When dissecting the results domain-wise, entropy consistently emerges as the favored metric in Relational Inference across all tested models, whereas the Planning domain shows a predominant preference for agreement for all but one model (GPT-3.5-turbo), as shown in Tables 4-7 in the Appendix.

Synthesizing these findings, agreement is the most effective consistency metric for Codex and most open-source models, closely followed by FSD. For GPT-3.5-turbo and GPT-4, FSD and entropy are closely matched in effectiveness. A conjectured reason for this discrepancy could be the lack of Reinforcement Learning from Human Feedback (RLHF) in Codex and open-source models, unlike GPT-3.5-turbo and GPT-4. We will revisit this by comparing a pair of models that only differ in the exposure to RLHF in Section 6.2.

Takeaways. Our findings indicate that consistency metrics offer a more reliable measure of confidence than baselines. The selection of the exact metric is model-dependent though, and we offer practical guidelines in Section 8.

5.2 The Role of Explanations

Does the prompting strategy influence how well a model can be calibrated? Here, we compare **standard** prompting, where the model only predicts the answer, against four **explanation-based** prompting strategies (CoT, LtM, PoT, and FCoT), where the model produces a reasoning chain before the answer. Figure 3 shows the results for each prompting strategy averaged across consistency metrics.

Explanations make large models better-calibrated. When large LMs (all GPT-family models, llama-13B, and llama-70B) are prompted to generate any form of explanation before the answer, they exhibit a marked improvement in calibration error ($p < 0.05$). Among smaller models, mistral-7B and mistral-7B-it show the same pattern, whereas the rest do not. Overall, the benefit of explanations on calibration is especially evident in larger models, mirroring the observed correlation between accuracy and model size with explanations (Wei et al. 2022).

GPT models are best calibrated with FCoT, while most open-source models are best calibrated with CoT. The calibration efficacy of GPT models (Codex, GPT-3.5-turbo, GPT-4) and Mistral-7B-it is maximized through FCoT prompting, which interleaves NL and SL. Conversely, when it comes to the remaining open-source models, CoT in pure NL appears to be the most effective in enhancing calibration overall. This contrast underscores a potential difference in how these closed-source and open-source models process and benefit from prompts involving explanations.

Takeaways. Including explanations in prompts not only bolsters LMs' performance (see Table 2 in Appendix 5) but also makes them better-calibrated. This dual benefit suggests that the process of generating explanations potentially aids models in better processing and reasoning about the tasks at hand, leading to outputs more closely aligned with expectations.

6 Analysis

In this section, we examine how scaling, instruction-tuning, RLHF, and sample size affect LMs' calibration properties.

6.1 How Does Scaling Affect Calibration?

We study how an increase in model parameters impacts different consistency metrics. Figure 4 compares Brier Score across all reasoning strategies (standard, CoT, LtM, PoT, and FCoT) for all three consistency metrics (Entropy, Agreement, and FSD) for different sized LLaMA models (7B, 13B, and 70B), in order to understand the effect of scaling on calibration. We observe that the average Brier Score across datasets goes down for all consistency metrics as the model scales up; suggesting that *scaling supports calibration*. In other words, the larger the model, the better it is calibrated across the various tasks studied in this paper.

Moreover, we observe that for LLaMA-7B, all prompting strategies have a very similar Brier Score, especially with FSD- and agreement-based metrics (as seen from the left parts of Figure 4). As the model scales up to 70B, the gap increases (to the right of Figure 4) between standard prompting and explanation-based strategies (all others). This shows that *explanation improves calibration with scale* for most cases.

6.2 How Do Instruction-Tuning and RLHF Affect Calibration?

To analyze the effect of instruction-tuning, we compare the calibration properties of Mistral-7B and Olmo-7B with their instruction-tuned versions (Mistral-7B-it and Olmo-7B-it)

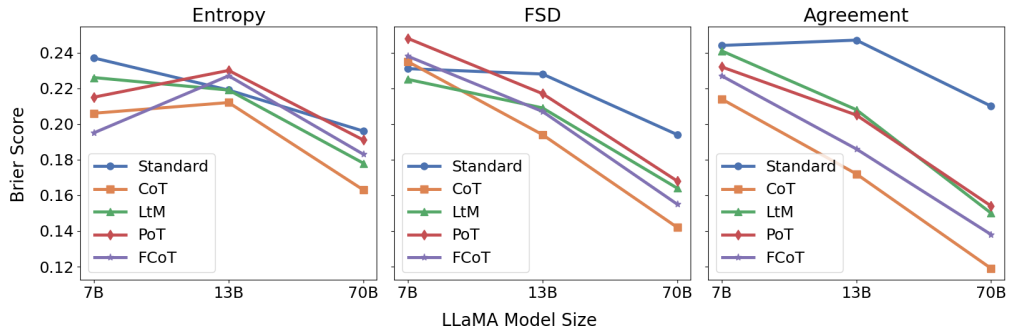

Figure 4: The Brier Score (↓) tends to improve as the model size increases for the 3 studied calibration metrics across most of the prompting techniques we consider.

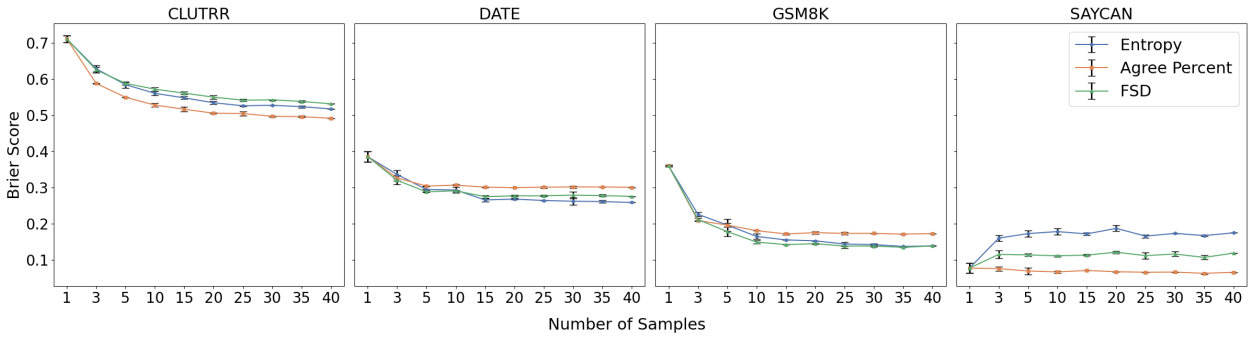

Figure 5: Brier Scores (↓) improve as we increase the number of samples for 3 of the 4 datasets. Results are obtained with GPT-3.5-turbo and CoT prompting. Each experiment was repeated five times, with the corresponding mean and standard variance reported.

across the four tasks we studied. Table 2 demonstrates that in general *instruction-tuning leads to worse calibration properties* for both Mistral and Olmo models. Additionally, the Olmo instruction-tuned model was further trained with RLHF policies (Bai et al. 2022) using DPO (Rafailov et al. 2024), resulting in Olmo-7B-it-rl. Our results reveal that *RLHF over instruction-tuning results in worse or unchanged calibration properties*, depending on the prompting strategy. Our findings are similar to those in reported in previous studies (Kadavath et al. 2022). One point to note is faithful prompting strategies improve calibration for Mistral-7B-it, and NL-based explanation (CoT, LtM) improves calibration for Olmo-7B-it and Olmo-7B-it-rl. This improvement could be attributed to variations in the instruction-tuning process, though it is difficult to pinpoint the exact cause.

6.3 How does the Number of Generated Outputs impact Calibration?

We analyze the usefulness of consistency-based calibration by generating different numbers of output samples and calculating different consistency metrics over them. Figure 5 demonstrates that generating *more samples can lead to better calibration scores, but the effect plateaus rather quickly* (with the SayCan dataset as an outlier, potentially associated with its low level of difficulty). We observe the improvement in the Brier Score as a function of the number of samples and the decision of the appropriate number of samples can be made based on the available computational budget and the desired calibration properties. Brier Scores usually saturate after $15-20$ samples, with a sharp drop at the beginning. With budget constraints, $3-5$ samples can already provide much more reliable confidence estimates compared to only sampling once.

7 Case Study: Does Calibration Help Improve Model Performance?

Beyond calibrating trust in model predictions, can consistency metrics contribute to improving task performance? To explore this, we perform a case study with GPT-3.5-turbo and GPT-4 on GSM8K and CLUTRR datasets from the MWP and Relation Reasoning domains respectively. We compare the consistency metrics against other calibration baselines in two experiments: **discriminating prediction correctness** and **improving final answer accuracy**.

In the first experiment, given a model's predictions $\hat{Y}$ on a dataset X, our goal is to differentiate the correctness of each prediction $\hat{y}_i$ with the confidence $\texttt{conf}(x_i, \hat{y}_i)$ provided by any calibration method. Identifying incorrect predictions is the first step for performance improvement, and it can be integrated into any self-correction pipeline (Madaan et al. 2023; Shridhar et al. 2023, i.a.). To test discrimination efficacy, we tune an optimal threshold θ for each calibration method on a development set.[5] If the provided confidence

[5] See Appendix 2 for tuning details.

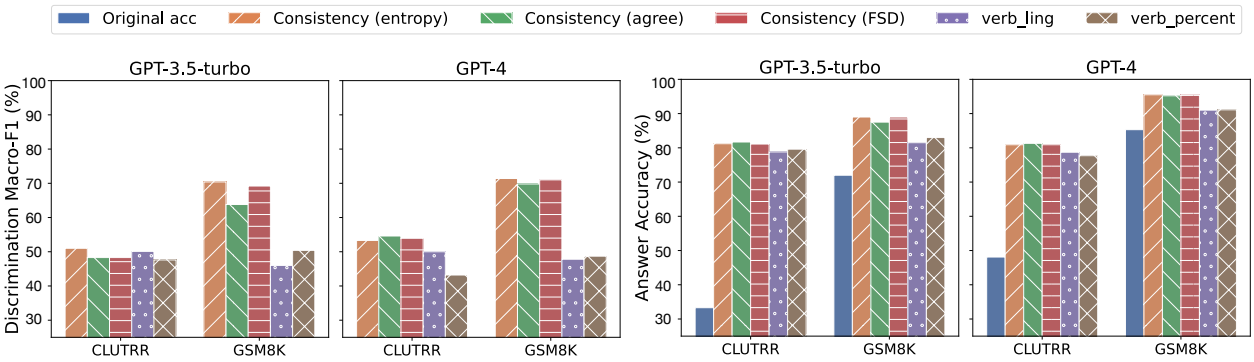

Figure 6: Left: Consistency-based calibration outperforms verbalized baselines in discriminating the correctness of predictions measured by Macro-F1 (↑). Right: Consistency-based calibration leads to a larger improvement in answer accuracy (↑) after correcting the top-k% most uncertain predictions with oracle answers. Scores are averaged across all prompting strategies.

score $\texttt{conf}(x_i, \hat{y}_i)$ is above θ, we consider the model prediction as correct, otherwise incorrect. Then, we evaluate the discrimination performance of each calibration method on the test set. The results, illustrated in Figure 6 (left), indicate that consistency metrics significantly outstrip verbalized baselines in discriminating correct and incorrect predictions, with the effect being most pronounced on the GSM8K dataset (more than doubled Macro-F1).

All three consistency metrics share this trend, except for the only case of GPT-3.5-turbo on CLUTRR, where entropy outperforms the optimal baseline, yet the gap between all methods is small.

Next, we assess the impact of calibration methods on answer accuracy, assuming subsequent self-correction steps are oracle. We choose this setting as discrimination is found to be the key bottleneck in the self-correction pipeline; once solved, LMs' are able to self-correct (Huang et al. 2023). In this experiment, we isolate the discrimination step in the pipeline and measure the impact of different calibration methods on the final accuracy. Given a model's predictions $\hat{Y}$ on a dataset X, we identify the top-k% most uncertain predictions, $\hat{Y}_{-}$, which are those with the lowest confidence scores according to the calibration method, as incorrect. This fixed k is chosen to be the true error rate of all model predictions, i.e., $k = 1 - \text{acc}(\hat{Y}, X)$. Finally, we correct $\hat{Y}_{-}$ with the ground-truth answers and evaluate the resulting accuracy. As shown in Figure 6 (right), post-correction accuracy exceeds original accuracy to the greatest extent when applying consistency-based calibration.

In both experiments, entropy and FSD are equally effective on GSM8K for both models, while agreement and entropy lead on CLUTRR for each model. In summary, consistency provides not just a measure of prediction trust, but can also contribute to enhanced model performance assuming ideal self-correction mechanisms.

8 Which Consistency Metrics Should I Use to Best Calibrate My Model?

Depending on a model's specific characteristics, such as its exposure to instruction-tuning and RLHF, we provide tailored recommendations for selecting appropriate consistency metrics for calibration (a prescriptive flowchart can be found in the extended version of this paper linked in the Abstract). Specifically, if the model has undergone both instruction-tuning and RLHF, either FSD-based or entropy-based consistency is a good starting point. Conversely, if the model has only been instruction-tuned without RLHF, agreement-based consistency is more suitable, with FSD as a good second choice. Finally, if the model has undergone neither instruction-tuning nor RLHF, agreement is recommended.

These suggestions are based on insights derived from our analyses in Sections 5 and 6. However, it is important to note that our research examined calibration properties in a somewhat limited scope, focusing on only four reasoning tasks across nine datasets. Additionally, certain comparisons (such as between RLHF and non-RLHF) are based solely on one pair of models (Olmo-7B vs. Olmo-7B-it). Consequently, our recommendations might not be universally applicable and should be applied judiciously.

9 Conclusion

We investigate the effectiveness of eliciting confidence in LLMs from sample consistency, using entropy and FSD as extensions of the naive agreement-based consistency measure. Through extensive evaluations on various open- and closed-source models and nine reasoning datasets, we demonstrate the superiority of these methods over traditional post-hoc verbalized and probabilistic calibration techniques. Further analysis shows that explanation generation, model scaling, and larger sample sizes improve calibration, while instruction-tuning has a counter-effect. In addition to providing more reliable confidence estimates, consistency measures also contribute to improved model performance assuming oracle subsequent self-correction steps. Finally, we provide guidance for selecting the most appropriate consistency metric for calibration based on different model types, sizes, and inference tasks, paving the way for more reliable and trustworthy applications of LLMs in various domains.

Acknowledgements

The research mentioned in this report is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200005 and Defense Advanced Research Projects Agency's (DARPA) SciFy program (Agreement No. HR00112520300). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or views, either expressed or implied, of ODNI, IARPA, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Brier, G. W. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1): 1–3.

Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; Julian, R.; et al. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, 287–318. PMLR.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *ArXiv preprint*, abs/2211.12588.

Chen, Y.; Yuan, L.; Cui, G.; Liu, Z.; and Ji, H. 2023. A Close Look into the Calibration of Pre-trained Language Models. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1343–1367. Toronto, Canada: Association for Computational Linguistics.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems.

Gal, Y. 2016. Uncertainty in Deep Learning.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Balcan, M.; and Weinberger, K. Q., eds., *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 1050–1059. JMLR.org.

Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, 10764–10799. PMLR.

Geng, J.; Cai, F.; Wang, Y.; Koeppl, H.; Nakov, P.; and Gurevych, I. 2023. A Survey of Language Model Confidence Estimation and Calibration.

Geva, M.; Khashabi, D.; Segal, E.; Khot, T.; Roth, D.; and Berant, J. 2021. *Did Aristotle Use a Laptop?* A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9: 346–361.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, 1321–1330. PMLR.

Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.; Song, X.; and Zhou, D. 2023. Large Language Models Cannot Self-Correct Reasoning Yet. In *The Twelfth International Conference on Learning Representations*.

Jiang, Z.; Araki, J.; Ding, H.; and Neubig, G. 2021. 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.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; Johnston, S.; El-Showk, S.; Jones, A.; Elhage, N.; Hume, T.; Chen, A.; Bai, Y.; Bowman, S.; Fort, S.; Ganguli, D.; Hernandez, D.; Jacobson, J.; Kernion, J.; Kravec, S.; Lovitt, L.; Ndousse, K.; Olsson, C.; Ringer, S.; Amodei, D.; Brown, T.; Clark, J.; Joseph, N.; Mann, B.; McCandlish, S.; Olah, C.; and Kaplan, J. 2022. Language Models (Mostly) Know What They Know. arXiv:2207.05221.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6402–6413.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing*

Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, 7167–7177.

Lin, S.; Hilton, J.; and Evans, O. 2022. Teaching Models to Express Their Uncertainty in Words. *Transactions on Machine Learning Research*.

Lyu, Q.; Havaldar, S.; Stein, A.; Zhang, L.; Rao, D.; Wong, E.; Apidianaki, M.; and Callison-Burch, C. 2023. Faithful chain-of-thought reasoning. *ArXiv preprint*, abs/2301.13379.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Welleck, S.; Majumder, B. P.; Gupta, S.; Yazdanbakhsh, A.; and Clark, P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *ArXiv preprint*, abs/2303.17651.

Manakul, P.; Liusie, A.; and Gales, M. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 9004–9017. Singapore: Association for Computational Linguistics.

Miao, S.-y.; Liang, C.-C.; and Su, K.-Y. 2020. A Diverse Corpus for Evaluating and Developing English Math Word Problem Solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 975–984. Online: Association for Computational Linguistics.

Mielke, S. J.; Szlam, A.; Dinan, E.; and Boureau, Y.-L. 2022. Reducing Conversational Agents' Overconfidence Through Linguistic Calibration. *Transactions of the Association for Computational Linguistics*, 10: 857–872.

Nye, M. I.; Tessler, M. H.; Tenenbaum, J. B.; and Lake, B. M. 2021. Improving Coherence and Consistency in Neural Sequence Models with Dual-System, Neuro-Symbolic Reasoning. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 25192–25204.

Papadopoulos, G.; Edwards, P. J.; and Murray, A. F. 2001. Confidence estimation methods for neural networks: A practical comparison. *IEEE transactions on neural networks*, 12(6): 1278–1287.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2080–2094. Online: Association for Computational Linguistics.

Portillo Wightman, G.; Delucia, A.; and Dredze, M. 2023. Strength in Numbers: Estimating Confidence of Large Language Models by Prompt Agreement. In Ovalle, A.; Chang, K.-W.; Mehrabi, N.; Pruksachatkun, Y.; Galystan, A.; Dhamala, J.; Verma, A.; Cao, T.; Kumar, A.; and Gupta, R., eds., *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, 326–362. Toronto, Canada: Association for Computational Linguistics.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Roy, S.; and Roth, D. 2015. Solving General Arithmetic Word Problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1743–1752. Lisbon, Portugal: Association for Computational Linguistics.

Shridhar, K.; Jhamtani, H.; Fang, H.; Van Durme, B.; Eisner, J.; and Xia, P. 2023. Screws: A modular framework for reasoning with revisions. *ArXiv preprint*, abs/2309.13075.

Shridhar, K.; Macina, J.; El-Assady, M.; Sinha, T.; Kapur, M.; and Sachan, M. 2022. Automatic Generation of Socratic Subquestions for Teaching Math Word Problems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 4136–4149. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Sinha, K.; Sodhani, S.; Dong, J.; Pineau, J.; and Hamilton, W. L. 2019. CLUTRR: A Diagnostic Benchmark for Inductive Reasoning from Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4506–4515. Hong Kong, China: Association for Computational Linguistics.

Srivastava, A.; Rastogi, A.; Rao, A.; Shoeb, A. A. M.; Abid, A.; Fisch, A.; Brown, A. R.; Santoro, A.; Gupta, A.; Garriga-Alonso, A.; et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Tam, D.; Mascarenhas, A.; Zhang, S.; Kwan, S.; Bansal, M.; and Raffel, C. 2022. Evaluating the factual consistency of large language models through summarization. *ArXiv preprint*, abs/2211.08412.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; brian ichter; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

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*, abs/2306.13063.

Yoo, K.; Kim, J.; Jang, J.; and Kwak, N. 2022. Detection of Word Adversarial Examples in Text Classification: Benchmark and Baseline via Robust Density Estimation.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q. V.; and Chi, E. H. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *The Eleventh International Conference on Learning Representations*.