Fact-and-Reflection (FaR) Improves Confidence Calibration of Large Language Models

Xinran Zhao[1,2,*] **Hongming Zhang**[1] **Xiaoman Pan**[1] **Wenlin Yao**[1]
Dong Yu[1] **Tongshuang Wu**[2] **Jianshu Chen**[1]
[1]Tencent AI Lab, Bellevue, [2]Carnegie Mellon University

Abstract

For a LLM to be trustworthy, its confidence level should be *well calibrated* with its actual performance. While it is now common sense that LLM performances are greatly impacted by prompts, the confidence calibration in prompting LLMs has yet to be thoroughly explored. In this paper, we explore how different prompting strategies influence LLM confidence calibration and how it could be improved. We conduct extensive experiments on six prompting methods in the question-answering context and we observe that, while these methods help improve the expected LLM calibration, they also trigger LLMs to be *over-confident* when responding to some instances. Inspired by human cognition, we propose Fact-and-Reflection (FaR) prompting, which improves the LLM calibration in two steps. First, FaR elicits the known "facts" that are relevant to the input prompt from the LLM. And then it asks the model to "reflect" over them to generate the final answer. Experiments show that FaR prompting achieves significantly better calibration; it lowers the Expected Calibration Error by 23.5% on our multi-purpose QA tasks. Notably, FaR prompting even elicits the capability of verbally *expressing concerns* in less confident scenarios, which helps trigger retrieval augmentation for solving these harder instances.

1 Introduction

With the emergence of Large Language Models (LLMs) (Chowdhery et al., 2022; Thoppilan et al., 2022; OpenAI, 2022, 2023; Touvron et al., 2023; Anil et al., 2023), various prompting strategies have been proposed for *improving the LLM performance*. It is now common sense that well-designed prompts can help elicit desirable capabilities from LLMs that are important for various tasks. For example, Chain-of-Thought and its variants have been widely used for unlocking LLM reasoning capabilities (Liu et al., 2022; Wei et al., 2022b; Wang et al., 2023; Press et al., 2023; Yao et al., 2023).

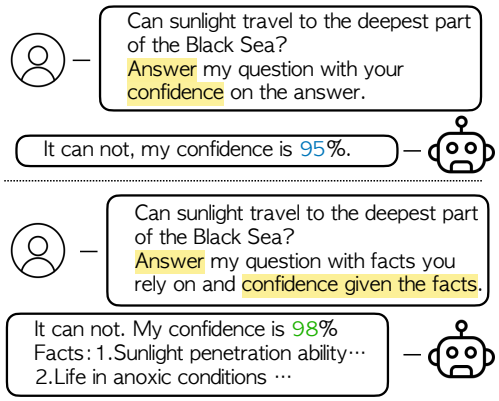

Figure 1: An example of how different prompting methods affect confidence extraction: model verbalized confidence of GPT-3.5 increases when asking the model to elicit the facts it relies on to answer the question.

However, what is less studied is the fact that prompting also influences the model confidence in their responses. For example, as shown in Figure 1, the LLM's verbalized confidence (Lin et al., 2022) would shift when it is asked to simply first seek supporting facts. Effective *confidence calibration* of LLMs — output confidence scores matching model performance — is crucial because calibrated confidence ensures the model reliability and helps guide practical use cases, e.g., identifying potential hallucinations (Kadavath et al., 2022; Varshney et al., 2023), applying additional fact-checking (Chen et al., 2021), etc. Beyond inference time usage, it can also help guide the training process, e.g., improving instruction tuning (Chung et al., 2022). Therefore, one central question we want to answer is: *how do different prompting methods influence the confidence calibration?*

To do so, we first start by assessing six different

[*] Work done during an internship at Tencent AI Lab, Bellevue. Corresponding contact email addresses: {xinranz3,sherryw}@andrew.cmu.edu, {hongmingzhang, xiaomanpan ,wenlinyao,dyu,jianshuchen}@global.tencent.com. Our code is publicly available at: https://github.com/colinzhaoust/fact-and-reflection.

Findings of the Association for Computational Linguistics: ACL 2024, pages 8702–8718
August 11-16, 2024

prompting strategies, on Question-Answering (QA) datasets that rely on reasoning (Geva et al., 2021) and knowledge (Bordes et al., 2014). In our assessment, we employ *confidence calibration*, specifically Expected Calibration Error (ECE) (Guo et al., 2017) and Macro Calibration Error (MacroCE) (Si et al., 2022), to measure the discrepancy between a model's performance and its confidence levels. They are widely recognized metrics for evaluating the quality of confidence scores in models, as discussed in prior studies (Desai and Durrett, 2020; Si et al., 2023). We find that different prompting methods generally suffer from *over-confidence*, and exhibit poor calibration at the instance level.

Psychological and cognitive research (Block and Harper, 1991; George et al., 2000) indicates that human's over-confidence can be mitigated by disentangling the processes of fact acquisition and reasoning. Instead of reasoning while stating the facts, explicitly recalling all relevant facts before deliberation can avoid early *anchoring* the reasoning process onto the first upcoming fact in the context. Accordingly, we propose our **F**act-**a**nd-**R**eflection (FaR) prompting to improve model confidence calibration — see Figure 2 for an example. Specifically, it consists of three steps: first, prompting the model for potential known facts and their sources, second, eliciting reflective reasoning to connect all recalled knowledge, and finally, generating the answer.

Our experiments on aforementioned datasets demonstrate that FaR prompting significantly reduces the confidence calibration error across various calibration measures (23.5% and 13.9% under ECE and MacroCE, respectively).

Further analysis reveals that the improvement comes from that FaR prompting intrigues the model to generate cautious answers that express concerns, such as adding a comment like "there is no sufficient evidence" after the answer. We observe that *expressing concerns* co-occurs with an average of 13.2% reduction in confidence relative to the situations without *expressing concerns*. This phenomenon can suggest a potential mechanism that helps detect hard instances that are not answerable with LLM's internal knowledge and may benefit from retrieval augmented generation.

In summary, our main contributions are:

1. We study how different prompting methods influence confidence calibration and find that many methods, though generally being helpful, suffer from the over-confidence issue.

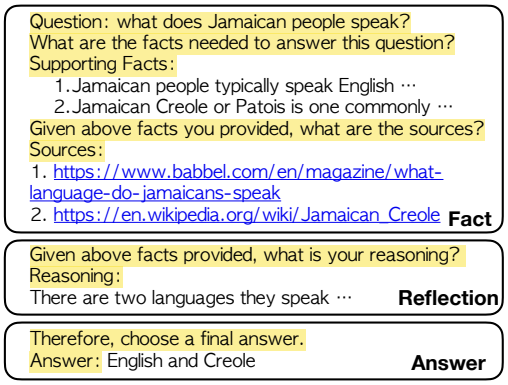

Figure 2: An example of the proposed FaR prompting, which consists of three steps: fact, reflection, and answer. Before answering the question and extracting the confidence scores, FaR explicitly decomposes the entire process into fact elicitation (with corresponding sources) and reflectively reasoning over the facts. The answers will then be utilized in the prompt to generate the final answer. Highlighted text denotes the input prompts provided to the model.

2. We propose FaR prompting that improves model confidence calibration across various common metrics.
3. We show that FaR prompting mitigates *over-confidence*. Moreover, FaR prompting elicits the model to express concerns when answering questions they are uncertain of.

2 How do Prompting Strategies Influence Confidence Calibration in LLMs?

To examine the influence of different prompting strategies on the confidence calibration of LLMs, we follow Tian et al. (2023) and conduct our experiments on QA. Specifically, we use QA datasets that mainly examine either reasoning (StrategyQA, Geva et al., 2021) or knowledge (Web Questions. Bordes et al., 2014) where the confidence calibration is important due to the reliability requirement. An example question would be, "Did Aristotle use a laptop?". The answers are either in the format of Yes/No or a set of short phrases, e.g., "English and Creole". We report the macro-average scores for two kinds of QA data. Following Wei et al. (2022a), we conduct all of our experiments using OpenAI's GPT-3 (`text-davinci-003`, Ouyang et al., 2022)[1]. We set the default max output length to 120 and the temperature to 1.2.

[1]At the time of writing, this model was the only one that both has sufficient reasoning capability for solving StrategyQA and provides access to the logits (necessary for confidence calculation). Experiments with other models are in Section 3.4.

2.1 Evaluation Metrics

We follow the conventional approach (Guo et al., 2017; Desai and Durrett, 2020; Si et al., 2022), and measure the confidence score quality with *confidence calibration*: the error between *model performance* and *confidence scores* across instances. Lower errors indicate better calibration and thus, higher accuracy in confidence scores.

We measure **model performance** with the exact match criterion, where an answer is deemed correct if it matches a candidate label after normalization.

Then, following Si et al. (2022), we use both Expected Calibration Error (ECE, Guo et al., 2017) and Macro-average Calibration Error (MacroCE, Si et al., 2022) to evaluate the quality of **confidence calibration**. Both metrics are based on the difference between the confidence scores and the correctness of model predictions.

- `ECE` uses a bucketing approach that measures the *overall calibration*. It assigns examples with similar confidence to the same buckets. Given the input x, ground truth y, prediction $\tilde{y}$, and B_m denoting the m-th bucket for (x,y,$\tilde{y}$), for N model predictions bucketed into M buckets:

$$\text{ECE} = \frac{1}{N}\sum_{m=1}^{M} |B_m| \cdot |\text{Acc}(B_m) - \text{Conf}(B_m)|,$$

  where $\text{Acc}(B_m)$ and $\text{Conf}(B_m)$ denote the accuracy and averaged confidence for the samples in B_m, and $|B_m|$ denotes the cardinality of B_m. Such definition triggers *bucket-canceling effect*, i.e., the over- and under- confident instances within the same bucket may cancel with each other and hence not contribute to the overall error. As a result, ECE provides **a stable overall measure** of how well the confidence matches the expected accuracy — A single outlier (e.g., extremely high confidence for a wrong prediction) will not affect the global ECE too much.
- `MacroCE` is the (macro) average of the following two instance-level calibration errors (ICE), which measures the ICE for the N_p correctly and N_n incorrectly predicted samples, respectively:

$$\text{ICE}_{\text{pos}} = \frac{1}{N_p}\sum_{i=1}^{N_p}(1 - \text{Conf}(x_i, \tilde{y}_i)), \forall_i y_i = \tilde{y}_i,$$

$$\text{ICE}_{\text{neg}} = \frac{1}{N_n}\sum_{i=1}^{N_n}(\text{Conf}(x_i, \tilde{y}_i) - 0), \forall_i y_i \neq \tilde{y}_i.$$

MacroCE does not have the bucket-canceling effects, provides more granularity, and is more reflective of **instance-level confidence calibration**. For example, consider a set of two predictions $\{1, 0\}$ with labels $\{0, 0\}$. When the corresponding confidence scores are $\{1, 0\}$, assume they are in the same bucket, the output of ECE will be 0, as the difference between the average confidence scores and accuracy. Therefore, the high error extreme instances are not captured by ECE (e.g., wrong, but conf $= 1$). In contrast, each instance contributes to MacroCE (equals 1 in this case), which reveals more instance-level effects than ECE.

2.2 Prompting Methods

We examine how different prompting methods influence confidence calibration. Besides *Standard* prompting, we categorize the strategies into two kinds: Step Decomposition and Multi-Candidate Selection (examples in Figure 7 in the appendix).

Step Decomposition prompts guide the model to output multiple steps of thoughts or knowledge that may facilitate answering the question in the final step. We consider three common ones:

1. `Knowledge prompting` (Liu et al., 2022): We insert a prompt "Generate some knowledge about the question:" after the original question, and have the model generate the final answer based on the generated knowledge. We also explore a minor variant: *Knowledge+Explain*, which denotes changing the final prompt from "Answer:" to "Explain and Answer:".
2. `Chain-of-Thought` prompting (CoT, Wei et al., 2022b): We have the model generate chained reasoning before the final answer. We follow (Kojima et al., 2023) to conduct zero-shot CoT by adding a prompt "Let's think step by step:" after the original question.
3. `Self-ask` (Press et al., 2023): we have the model decompose the original question into multiple sub-questions, generate the intermediate answers, and combine them in the prompt to get the final answer. In detail, we first ask the model "Are follow-up questions needed?". If not, we directly get the final answer; Otherwise, we ask the model to generate the follow-up questions and the corresponding intermediate answers. All these intermediate steps are included in the final prompt ("Question:<question>; Intermediate Questions and Answers: <generated question-answer pairs> Answer:"). *Self-Ask (ag-*

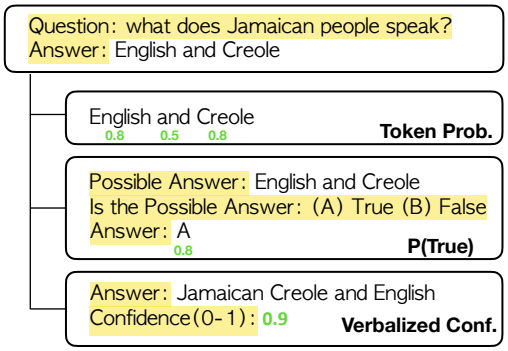

Figure 3: Different methods for computing confidence scores. The numbers in green denote the token probability of the output sequences. Highlighted texts denote input prompts provided to the model. Note that Both P(True) and Verbalized Confidence can be considered as suffix prompting methods that obtain the confidence scores after the model outputs its answer.

gregate) denotes the variant that asks the model to answer all the intermediate questions together in a one-step generation.

On the other hand, **Multi-candidate Selection** methods acquire the final answer through querying the model for multiple rounds and selecting a candidate with pseudo discussion or majority voting. We examine the following two (we extract their confidence scores after the final answer):

4. `Self-consistency` (Wang et al., 2023): We have the model generate the answers multiple times with the standard prompt, and select the final answer through a majority vote. We follow the original work to sample 10 different outputs with a temperature 0.7.
5. `Tree-of-Thought` prompting (ToT, Yao et al., 2023): We follow the naming protocol by Hulbert (2023), and have the model to perform state-aware generation and search, by mimicking multiple experts discussing the input questions in pseudo conversations. We denote our method as pseudo-ToT since the search steps are not conducted with a separate module.

2.3 Confidence Extraction Methods

To obtain **confidence scores**, we examine three widely used methods in literatures (Figure 3):

- `Token Prob.`: Computed by averaging the top-1 log probability of each token over the entire sequence, and then applying the exponential. This is the reciprocal of the perplexity of the generated sequence with greedy decoding.
- `P(True)` (Kadavath et al., 2022): After prompting the model to generate the answer with "*Possible answer: <model_answer>*", it appends a suffix prompt asking "*Is the possible answer: (A) True (B) False*", and then extracts the probability of answering "*A*" as the confidence score.
- `Verbalized Confidence` (Lin et al., 2022): After the model generates the answer, it further prompts the model to generate the confidence score by using suffix "*Confidence (0-1):*"[2].

Conf. Extraction	ECE ↓	MacroCE ↓
Token Prob.	27.3	80.8
P(True)	41.3	70.3
Verbalized	42.7	106.3

Table 1: Expected Calibration Error (ECE) and Macro-average Calibration Error (MacroCE) of different confidence extraction methods in Table 2. Results are averaged over different prompting methods. Down arrows indicate the lower is the better.

To validate how applicable these methods are in our scenario, we first compare different confidence extraction methods to determine the ones to be used in the later experiments. We use the ECE and MacroCE metrics averaged over different prompting methods to measure the overall and instance-level performance (the lower, the better). The average is calculated over the 8 baseline methods in Table 2.

Table 1 shows that overall, Token Prob. achieves the best ECE scores. On the other hand, P(True) achieves the best MacroCE scores. The reason can be that, while P(True) consistently measures the probability of an option for a multiple-choice question, Token Prob. can be unstable when the generated answer is long and includes auxiliary words ("English and Creole" vs. "Jamaican people speak English and Creole" in Figure 3). Verbalized Confidence shows the worst performance among the confidence extraction methods, indicating that further improvement is still required (e.g., tailor-made instructions shown in Tian et al. (2023)). Therefore, we exclude Verbalized Confidence extraction method in our following experiments, and for simplicity, we only report an aggregated overall performance of calibration by averaging the metrics obtained from using Token Prob. and P(True)[3].

[2]Slightly different from Lin et al. (2022), we further add the hint for the range of confidence score "(0-1)".

[3]See Table 8 in the appendix for the full results. Since the confidence scores are on the same scale (0-1) and a specific calibration metric (e.g., ECE) essentially measures the same kinds of error regardless of the confidence extraction methods, reporting the average measures the overall performance.

Prompting Method	ECE ↓	MacroCE ↓
Standard	30.3	**54.6**
Step Decomposition		
Knowledge	33.0	73.9
Knowledge+Explain	27.1	64.5
CoT	29.6	62.3
Self-Ask	26.4	66.6
Self-Ask (aggregate)	**26.0**	66.0
Multi-Candidate Selection		
Self-Con.	34.7	67.6
Pseudo-ToT	33.0	73.5

Table 2: Expected Calibration Error (ECE) and Macro-average Calibration Error (MacroCE) of different prompting methods of different categories, as introduced in Section 2.2. The down arrow implies the lower, the better. The best-performing entry on each column is marked in **bold**.

2.4 Impact of Prompting Methods

Table 2 shows the impact of prompting methods on confidence calibration. We observe:

Step Decomposition helps improve the model's overall expected confidence calibration (Knowledge+Explain, CoT, Self-Ask), as reflected by ECE in Table 2. The reason can be that, with the generated chained thoughts or intermediate question-answer pairs in the context, the model-generated confidence is grounded onto the elicited thoughts.

Multi-Candidate Selection may cast a negative effect on the confidence calibration (Self-Con., Pseudo-ToT). The reason can be that the candidate proposal stage adds to the randomness of the context. The final answers are sampled from and not necessarily aware of other candidates when generated, but the confidence extraction methods are conditioned on all candidates. Such mismatch can lead to lower confidence calibration and suggests that specialized confidence extraction methods should be developed to improve their performance.

Compared to the standard prompting, *the advanced prompting methods (e.g., CoT) seem to degrade the instance-level calibration* i.e., they can simultaneously achieve higher MacroCE but lower ECEs. Such mismatch suggests that instance-level extreme values have a negative impact on the confidence calibration. We then raise the question: what are the major causes of these high error extreme values: over-confidence (high confidence in wrong answers) or under-confidence (low confidence in correct answers)? We select the best-performing prompting methods to investigate further.

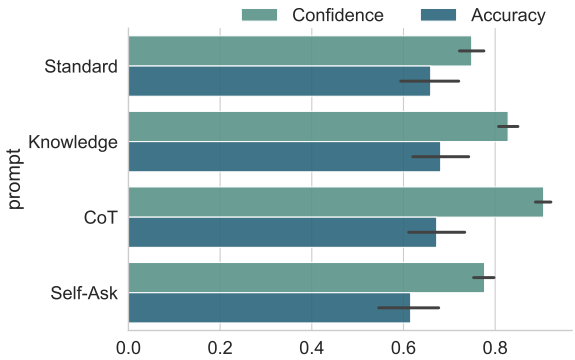

Figure 4: Accuracy versus confidence scores. Higher confidence scores relative to accuracy mean that the model is generally over-confident. In the ideal calibration case, there should be no gap between them.

Over-Confidence Degrades Calibration We compare the averaged confidence scores (over all the samples) and accuracy in Figure 4. In the ideal case where we reach the perfect instance-level calibration (i.e., 100% confidence on all correct examples and 0% confidence on all wrong examples), the averaged confidence scores shall be equal to the accuracy of the model on the data.

We can observe that, compared to the standard prompting, *all other prompting methods suffer even more from over-confidence.* Similar to well-observed *anchoring bias* in human behavior (Tversky and Kahneman, 1974; Furnham and Boo, 2011; Lieder et al., 2018), the generated thoughts may also mislead the model to be over-certain of its own answer. For example, the generated thoughts such as *from the steps we can conclude that* can potentially make the model confident with its answers. If the output answer is wrong, the instance-level error from these high-confidence cases will be high. In the following section, we will investigate if treatment towards human *anchoring bias* can be extended to model confidence calibration.

3 Fact-and-Reflection Prompting

3.1 Motivation and Design

There have been studies in cognitive science on how to mitigate the over-confidence caused by anchoring-bias (Block and Harper, 1991; George et al., 2000), and the ideas have also been recently introduced to AI research in the context of decision-making (Echterhoff et al., 2022). One key insight is: *Providing multiple facts, instead of conducting reasoning directly with the first acquired fact, can help reduce the bias.* We take inspiration from this finding and propose to improve confidence cal-

ibration by *decomposing the fact-acquiring and reflective reasoning steps during prompting*. We expect this framework to encourage the inclusion of multi-perspective facts in the context before reasoning toward the answer. We denote it as **F**act-**a**nd-**R**eflection (**FaR**) prompting, which can be viewed as a new form of step decomposition prompting with constrained decomposition.

Specifically, in FaR prompt context, we guide the models with facts, sources of facts, and reflective reasoning conditioned on the facts before extracting confidence. In the CoT-style prompting, the answer A is sampled from the $p(A|Q, T, \theta)$ (Dohan et al., 2022), where Q denotes the query, T denotes the thoughts generated by the model and θ denotes the model parameters. In FaR, we sequentially acquire two component thoughts T_{f} and T_{r} (i.e., fact and reflection steps in Figure 2) regarding the known facts and reflective reasoning of the model about the questions. Motivated by Weller et al. (2023), we believe the trustworthiness of the sources can help stabilize the model reflection T_{r}. For model generated T_{f}, we add a sub-step to ask the model to elicit the known sources of the generated facts.

Upon acquiring the component thoughts, The final answer A is sampled from $p(A|Q, T_{\text{f}}, T_{\text{r}}, \theta)$, i.e., the model generates the answer with the final step prompt including thoughts at each step.

We design FaR prompting to be orthogonal to confidence extraction so that it can work together with different confidence extraction methods. The results of the step questions and the final answer will be utilized in the prompt to extract confidence.

In an ideal case, the model-generated "facts" should be verified by humans. However, human annotations are not always available and are hard to collect. To address this, we propose to prompt the models to generate relevant internal knowledge that may potentially help answer the questions as "facts". We conduct a pioneering study on introducing external verification in Section 3.3.

3.2 Performance and Ablations

We further study the generalized case with model-generated knowledge as T_{f}. We compare FaR and its different variants with the strongest baselines in Table 2 (i.e., entries with the lowest calibration errors measured in ECE and in MacroCE), which are Self-Ask (aggregate) and Standard, respectively. Table 3 shows that **FaR prompting brings improvements on both calibration metrics.**

Prompting Method	ECE ↓	MacroCE ↓
Standard	30.3	54.6
Self-Ask (aggregate)	26.0	66.0
FaR (fact-only,no-source)	26.4	72.3
FaR (fact-only)	29.5	70.5
FaR (no-source)	28.4	60.5
FaR (+*explain*)	27.4	71.7
FaR (final)	**22.8**	**47.0**

Table 3: Expected Calibration Error (ECE) and Macro-average Calibration Error (MacroCE) of different ablations of FaR. Standard and Self-Ask (aggregate) are provided for reference. "↓" denotes that lower is better. The best performance of each column is in **bold**.

We also include an extensive ablation study over the prompt components of FaR: *fact-only* denotes the variant that we do not conduct the reflection step; *no-source* is where we remove source generation (i.e., no "what are the sources..." step in Figure 2); *+explain* changes the final prompt from "Answer:" to "Explain and Answer:", which is motivated by *Knowledge+Explain*. From the table, we can observe that, although all variants achieve lower ECE than Standard Prompting, our final design (denoted as *FaR (final)*) achieves the best performance among them. On the other hand, for MacroCE, most variants get lower performance than the standard prompting. Therefore, all the components in FaR are important in achieving good performance in both metrics.

3.3 What Impacted the Calibration?

The performance with ECE and MacroCE in Table 2 and 3 together presents that FaR *does* mitigate the over-confidence issue analyzed in Section 2.4, with both the correct and wrong predictions considered in the design of MacroCE.

We now analyze *how* FaR prompting shifts the confidence distribution of the LLM generation. We compare it with CoT, as it is one of the best-performing models in the Step Decomposition category (Table 2). In addition, FaR can be regarded as a specially structured step decomposition prompting that explicitly disentangles the steps of knowledge self-prompting and reflective reasoning.

Figure 5 presents the specific distribution of confidence scores extracted by Token Prob., P(True), and Verbalized, respectively, smoothed by Kernel Density Estimation. We can observe that FaR **mitigates the overconfidence issue** that we aspire to

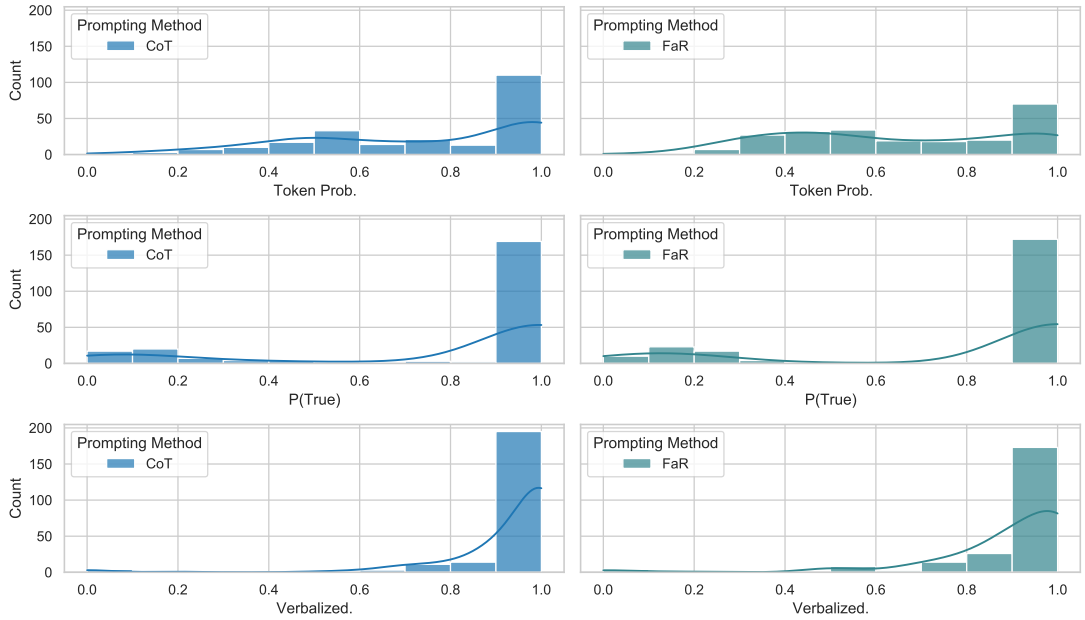

Figure 5: Distribution of the confidence scores (*Token Prob., P(True), Verbalized*, in the top, middle, bottom, respectively), with different prompting styles, CoT (left) and FaR (right). Kernel Density Estimation (curved lines) is used to show smoothed distributions. FaR prompting helps lower the distribution mass on the high confidence side for different confidence extraction methods.

resolve for different confidence extraction methods — it damps down the peak values and moves the density mass to the left side (lower confidence). That said, it still suffers from overconfidence. Such behavior partially explains the better confidence calibration results we observed in Section 2.4. Even the density mass of the verbalized confidence is much balanced with FaR prompting, which greatly enhances the usability of such a method for black-box language models that have no access to the output token probabilities. Again, consistent with our findings in Section 2.4, various confidence extraction methods still suffer from *over-confidence* issue, i.e., the confidence scores are higher than the target model performance, especially Verbalized Confidence. With Verbalized Confidence, even though FaR helps reduce over-confidence, further effort shall be made in future work to make it better calibrated.

Expressing Concern. How is FaR mitigating overconfidence? As shown in Table 4, qualitatively, we observe an interesting phenomenon of the model outputs, named *Expressing Concern*. With FaR prompting, besides outputting the answer, the model further specifies its thoughts on the answers, including (i) if the current evidence provided in the context is enough to answer the question; (ii) if a specific condition should be given to answer the question.

Question (Label)	Model Output
Would Persephone be a good consultant to a landscape architect? (True)	False. There will need to be further research.
Would an owl monkey enjoy a strawberry? (True)	It is not possible to answer with current evidence this question.
Does Post Malone have a fear of needles? (False)	False, but there is not yet sufficient evidence to answer.
Should a Celiac sufferer avoid spaghetti? (True)	False (It depends on the ingredients of the spaghetti)

Table 4: Examples of the model expressing concerns. In addition to the output answer, the model also specifies whether the knowledge is sufficient or whether further conditions are needed to make the prediction. See Table 9 for detailed outputs for other steps, e.g., diverse model-generated sources.

Comparing the probability of the model expressing concern using different prompts, FaR prompting can inspire the model to express concerns, compared to using the original CoT prompting (in 8.8% vs 3.9% examples). If we further remove the constraint on choosing one answer in the instruction, the model will elaborate further comments besides giving the answers (e.g., "False. there is not yet sufficient evidence") in 59.2% cases with FaR prompting (denoted as FaR(free)). These complex an-

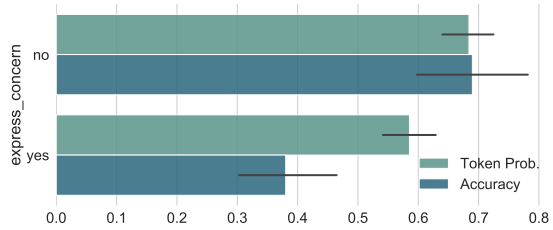

Figure 6: The confidence (extracted via token probability) and the accuracy when the model expresses (Lower) and does not express (Upper) concerns. With FaR (free) prompting, LLMs are more inclined to express concerns when the confidence and accuracy are low.

Prompting Method	ECE ↓	MacroCE ↓
Vicuna-13b		
Standard	19.4	35.5
FaR	**11.1**	**33.9**
Baichuan-2-13b		
Standard	17.1	64.3
FaR	**10.1**	**48.4**
Llama-2-13b		
Standard	19.3	71.9
FaR	**15.4**	**54.2**

Table 5: Expected Calibration Error (ECE) and Macro-average Calibration Error (MacroCE) of different prompting methods and different backbone large language models. The down arrow implies the lower, the better. The best-performing entry on each column is marked in **bold**.

swers are harder for performance computation than simple short answers, but give a clear picture for reviewing the model comments on the given questions. For example, as shown in Table 4, the model correctly answers that Post Malone does not have a fear of needles [4], but it still expresses concerns about insufficient evidence. Upon the concerns, users can choose to find further evidence not in the model training data to help verify the guess (e.g., some video interviews). Yet the definition of "good" concerns requires future investigation.

Notably, as shown in Figure 6, the expressing of concerns in the output typically co-occurs with lower confidence scores, which in turn co-occurs with lower accuracy on the questions. This implies that the model exhibits better confidence calibration — with FaR prompting, the model tends to express concern on **hard examples**. As a result, it can further act as a kind of AI feedback (Bai et al., 2022) for self-improvement by identifying difficult instances. For example, such a signal can trigger an iterative application of retrieval augmentation for checking and correcting the facts and sources provided by the models (Chen et al., 2021).

When extracting the statistics in Figure 6 from FaR(final) and CoT, on examples that the model expresses concerns, the accuracy is 25% (FaR) vs. 60.2% (CoT); on examples that models do not express concerns, the accuracy is 67.0% (FaR) vs. 69.9% (CoT). That is, FaR identifies examples with lower overall accuracy than CoT when concerns are expressed.

Addressing hard examples. We further simulate the scenario of detecting hard examples that are not answerable with LLM's internal knowledge and then using external knowledge as augmentation. We only sample and apply retrieval augmentation to the examples where the model expresses concerns. The treatment is done by adding corresponding external knowledge to the hard examples in the context, with the same setting and external knowledge from Zhao et al. (2023)[5].

We compare this sampling strategy with randomly sampling the same portion of examples (i.e., control) and check the performance gain. With FaR(final) as the prompting method, we can observe a 68.0% performance improvement in accuracy through external augmentation on hard examples that the model expresses concerns about. In contrast, randomly sampling the same portion of examples and applying augmentation only achieved a 15.0% performance improvement in accuracy, implying that FaR(final) identifies the instances demanding the external knowledge more accurately by checking if the model *expresses concerns*.

3.4 Generalizing to other language models

In this section, we generalize our experiments beyond GPT-3 to test the robustness of FaR across other language models. We follow the previous settings and conduct experiments on StrategyQA, with Token Prob. as the confidence extraction method. We extend our experiments with Vicuna (`Vicuna-13b-v1.3`, Zheng et al., 2023), Baichuan (`Baichuan2-13B-Chat`, Baichuan, 2023), LLama 2 (`Llama-2-13b-chat-hf`, Touvron et al., 2023). Our inference structure is built

[4]The model supports its educated guess with Post Malone's numerous tattoos in the fact step. See Table 9 for details of fact and reflection steps of this and other examples.

[5]Further analysis on this setting incorporating external knowledge is in the appendix (Section A.4) .

with vLLM (Kwon et al., 2023).

From Table 5, we can observe that, compared to Standard Prompting, FaR consistently leads to reduced calibration errors, measured by either ECE or MacroCE, which suggests the generalizability of FaR towards open-source language models.

4 Related Work

Prompting Large Language Models. Recent research (Brown et al., 2020; Kojima et al., 2023) on large language models shows that in-context learning (ICL) achieves great effectiveness in using models as few-shot or zero-shot reasoners. Different styles of prompting such as Knowledge prompting (Liu et al., 2022), Chain of Thought (CoT) prompting (Wei et al., 2022b), Self-Consistency prompting (Wang et al., 2023), Self-ask (Press et al., 2023), Tree-of-Thought prompting (Yao et al., 2023), and Skill-in-Context (Chen et al., 2023) are then proposed to guide the model to elicit its knowledge for reasoning in different ways.

Most previous work mainly focuses on how such a prompting method influences the model performance on various tasks. In this paper, we compare how confidence calibration is influenced by different prompting methods.

Confidence Calibration of LLMs. Extracting honest and constructive confidence scores of large language models is considered an important step towards building faithful and trustworthy AI systems (Desai and Durrett, 2020; Si et al., 2023). Many methods have been proposed recently to get reliable confidence scores with different suffix prompts added after outputting possible answers, such as a follow of True or False multiple choice question (Kadavath et al., 2022), asking models to describe the likelihood of the answer (Lin et al., 2022), and describing the likelihood that the answer is correct with demonstrations (Tian et al., 2023). However, it remains unclear how robust the methods are and how good they are comparatively. Our paper proposes FaR prompting as an orthogonal method to improve calibration and compare different extraction methods with our test bed.

Recently, Yang et al. (2023) discuss the honesty problem of models as part of the alignment. Qian et al. (2023) study the confidence change when there is a conflict between in-context and model internal knowledge. Another line of work links model confidence with human confidence (Zhou et al., 2023; Steyvers et al., 2024; Zhou et al., 2024). In our paper, we refer to the model trustworthiness based on confidence calibration.

5 Conclusion and Discussion

We closely examined how different prompting methods influence confidence calibration and found that over-confidence may lead to bad instance-level calibration. To address that, we propose FaR prompting, which decomposes the fact elicitation and reflective reasoning steps, and shows that it provides a good way to calibrate the model performance across different metrics. Our further analysis reveals that the reasons behind the performance can be that FaR prompting elicits the model to generate more *honest* answers, via *expressing concerns* in lower confidence situations.

We encourage the designers of future prompting methods to evaluate their influence on the confidence calibration in addition to the performance. Meanwhile, we further suggest that future confidence extraction methods should take into account their robustness to different prompting methods.

Limitations

Extension to Human Instruction Datasets. So far, we have conducted our experiments on question-answering datasets and shown the effectiveness of current prompting methods, confidence extraction methods, and our FaR prompting. Our method may be examined on the human instruction datasets as well, such as the Human Eval dataset (Wang et al., 2022). However, since the targeted answers are in free form (e.g., ranging from designing a personal profile to writing a program for quick sort), it may require very different evaluation metrics beyond simple accuracy, which calls for further study and is thus beyond the scope of this paper. Multi-perspective evaluation of the output can be necessary, such as the annotations provided in (Malaviya et al., 2023). We consider such extension to be an important future work.

Inner Model Dynamics. The confidence extraction methods compared in this paper are mainly based on signals from the model's logits or final sampled tokens. It remains unclear how prompting methods will influence the model uncertainty estimated from inner model dynamics, e.g., the tree structure-ness of a sentence inside transformers (Murty et al., 2023). We still study the LLM model as a black box and do not closely examine how the model's internal working mechanisms are

influenced by the FaR prompting. One future direction is to study the confidence calibration problem by using mechanistic interpretation (Olsson et al., 2022). We will leave such internal interpretability work as a future work.

Model Elicited Sources. In this paper, we explore how model-generated knowledge sources help improve model confidence calibration. In the Appendix, we present the detailed outputs for all steps of FaR, such as diverse model-generated sources that include but are not limited to URLs, in Table 9. For example, in the last row of Table 9, the model elicits the sources as *the Mayo Clinic and the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK)*. The model then autonomously comments on whether these sources are reputable, which plays the role of verbal confidence extraction on generated facts. However, Gao et al. (2023) directly generating sources based on facts may lead to inaccurate sources. Exploring the relation between the source's factualness and confidence calibration can be an important future direction.

Ethical Considerations

Our datasets are written by professional annotators or extracted from Wikipedia for the purpose of scientific research on question-answering systems. However, we observe no model outputs that use extremely sensational language or inappropriate and aggressive language. The questions and outputs collected are all in English, which can limit the generalizability of the performance of our pipeline.

Acknowledgments

This work is initiated during Xinran's internship at Tecent AI Lab, Bellevue. Xinran is supported by the ONR Award N000142312840. The authors thank Zhengxuan Wu, Xuanyu Zhou, Ruixin Hong, Vijay Viswanathan, Chenyang Yang, and Christina Ma for their insightful feedback and anonymous reviewers for helpful discussions and comments.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Richard A Block and David R Harper. 1991. Overconfidence in estimation: Testing the anchoring-and-adjustment hypothesis. *Organizational Behavior and Human Decision Processes*, 49(2):188–207.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings.

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

Jiaao Chen, Xiaoman Pan, Dian Yu, Kaiqiang Song, Xiaoyang Wang, Dong Yu, and Jianshu Chen. 2023. Skills-in-context prompting: Unlocking compositionality in large language models. *arXiv preprint arXiv:2308.00304*.

Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. 2021. Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A. Saurous, Jascha Narain Sohl-Dickstein, Kevin Murphy, and Charles Sutton. 2022. Language model cascades. *ArXiv*, abs/2207.10342.

Jessica Maria Echterhoff, Matin Yarmand, and Julian McAuley. 2022. Ai-moderated decision-making: Capturing and balancing anchoring bias in sequential decision tasks. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Adrian Furnham and Hua Chu Boo. 2011. A literature review of the anchoring effect. *The Journal of Socio-Economics*, 40(1):35–42.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations.

Joey F George, Kevin Duffy, and Manju Ahuja. 2000. Countering the anchoring and adjustment bias with decision support systems. *Decision Support Systems*, 29(2):195–206.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks.

Dave Hulbert. 2023. Tree of knowledge: Tok aka tree of knowledge dataset for large language models llm. `https://github.com/dave1010/tree-of-thought-prompting`.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Falk Lieder, Thomas L Griffiths, Quentin J M. Huys, and Noah D Goodman. 2018. The anchoring bias reflects rational use of cognitive resources. *Psychonomic bulletin & review*, 25:322–349.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169, Dublin, Ireland. Association for Computational Linguistics.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2023. Expertqa: Expert-curated questions and attributed answers. In *arXiv*.

Shikhar Murty, Pratyusha Sharma, Jacob Andreas, and Christopher D. Manning. 2023. Characterizing intrinsic compositionality in transformers with tree projections. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

OpenAI. 2022. Chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models.

Cheng Qian, Xinran Zhao, and Sherry Tongshuang Wu. 2023. "merge conflicts!" exploring the impacts of external distractors to parametric knowledge graphs.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2023. Prompting gpt-3 to be reliable. In *International Conference on Learning Representations (ICLR)*.

Chenglei Si, Chen Zhao, Sewon Min, and Jordan Boyd-Graber. 2022. Re-examining calibration: The case of question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2814–2829, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mark Steyvers, Heliodoro Tejeda Lemus, Aakriti Kumar, Catarina Belem, Sheer Karny, Xinyue Hu, Lukas Mayer, and Padhraic Smyth. 2024. The calibration gap between model and human confidence in large language models. *ArXiv*, abs/2401.13835.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *ArXiv*, abs/2305.14975.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou,

et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2023. " according to..." prompting language models improves quoting from pre-training data. *arXiv preprint arXiv:2305.13252*.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023. Alignment for honesty.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, and Jianshu Chen. 2023. Thrust: Adaptively propels large language models with external knowledge.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Kaitlyn Zhou, Jena D. Hwang, Xiang Ren, and Maarten Sap. 2024. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. *ArXiv*, abs/2401.06730.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models.

A Appendix

A.1 Examples of Baseline Prompting Methods

In Section 2.2, we provide a detailed description of all the baseline prompting methods. Here we further illustrate their working mechanisms in Figure 7. It shows how these methods are categorized into two major classes — Step Decomposition and Multi-Candidate Selection — based on the number of answer candidates are provided during the inference stage. All the baseline methods are evaluated on the original splits of StrategyQA (dev, 229 examples) and Web Questions (test, 100 examples), respectively. The data points we evaluated will be released upon acceptance.

A.2 Details about Over-confidence or Under-confidence

As we discussed earlier, most of the prompting methods generally suffer from over-confidence. We now include some additional results to further show that this is true across different confidence extraction methods.

In Figure 8, we show the average confidence scores and the accuracies of different prompting methods, where the confidence score for each method is obtained by averaging the confidence scores of all samples. From Figure 8, we can observe that our conclusion on over-confidence exists generally across different confidence extraction methods and prompting methods. We can also identify that Verbalized Confidence suffers most from the over-confidence issue.

A.3 Impact of Confidence Extraction Methods with Additional Metrics

Conf. Extraction Methods	ECE-avg ↓	ECE-wins ↑	MacroCE-avg ↓	MacroCE-wins ↑
Token Prob.	23.5	8	78.4	4
P(True)	37.2	1	65.4	5
Verbalized	40.6	0	101.9	0

Table 6: Expected Calibration Error (ECE) and Macro-average Calibration Error (MacroCE) of different confidence extraction methods. Results are averaged over different prompting methods. Up (Down) arrows indicate the higher (lower) is the better.

In Section 2.3, we compare different confidence extraction methods using the ECE and MacroCE (i.e., ECE-avg and MacroCE-avg) averaged over different prompting methods to measure the overall and instance-level performance (the lower, the better). To ensure that extreme values do not influence the conclusion, we further report the additional metric of ECE/MacroCE-wins in Table 6. If a confidence extraction method achieves the lowest ECE/MacroCE-based errors with respect to a prompting method, it is marked as a win. All the scores are the average of 8 baseline methods in Table 2 together with FaR (final).

From Table 6, we observe that Token Prob. achieves the best ECE for both the averaged scores and wins. In contrast, P(True) achieves the best MacroCE for both the averaged scores and wins, though the margin is not large. Therefore, the conclusion based on this new metric is consistent with the earlier findings.

Prompting Method	ECE ↓
Standard	25.5
FaR (human fact-only)	14.8
FaR (human fact + reflection)	**13.6**

Table 7: Expected Calibration Error (ECE) of using human-annotated facts in FaR, denoted by FaR (human). We use token probability as the confidence extraction method. The down arrow denotes that a lower score implies better calibration performance.

A.4 FaR with Human-Annotated External Knowledge

To further verify the idea that providing facts helps with confidence calibration, we incorporate multiple human-annotated facts (i.e., T_f) that are relevant to each question (Q) on the StrategyQA task (Geva et al., 2021). The benchmark in this experiment is not the mix of StrategyQA and WebQ since human-annotated supporting facts are only provided in StrategyQA. For example, for the question "Did Aristotle use a laptop", one piece of the fact can be "The first laptop was invented in 1980". Note that we assume the human-annotated facts are accurate, and therefore omit the step of generating the source. Incorporating human-annotated facts isolates the imperfection that may arise from model-generated facts and sources, which serves as a controlled ablation for examining the helpfulness of facts. As shown in Table 7, including human-annotated facts in the context can improve the confidence calibration significantly.

Furthermore, with the reflection step linking the facts and the final answer, the calibration can be further improved. However, human annotations are not always available and are hard to collect. To

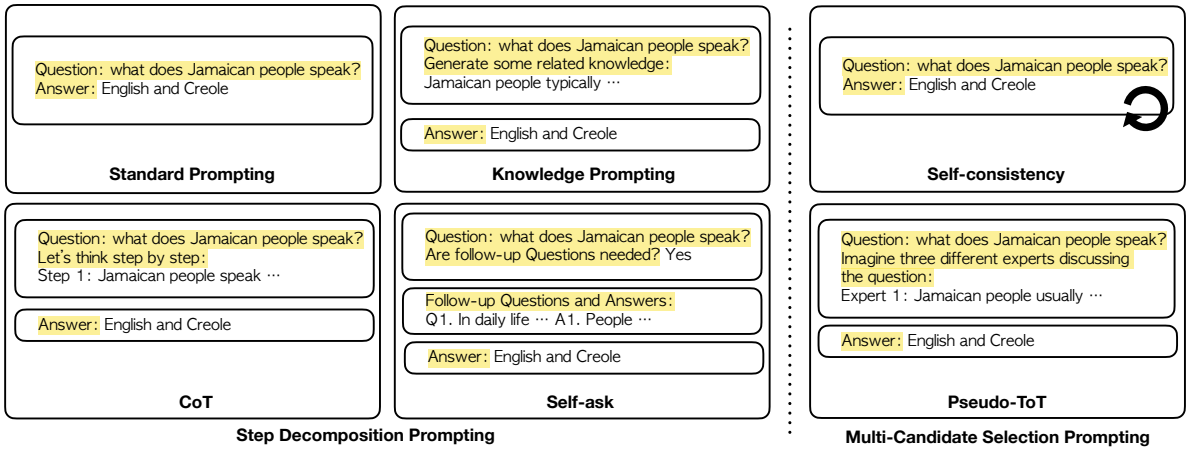

Figure 7: Different existing prompting methods to be analyzed. Each rounded rectangle represents one round of prompting. The intermediate answers will then be utilized in the final prompt to extract the answer to the original question. Highlighted text denotes the sentences provided to the model as the prompt. The loop icon in *self-consistency* prompting denotes re-sampling with the same prompt multiple times.

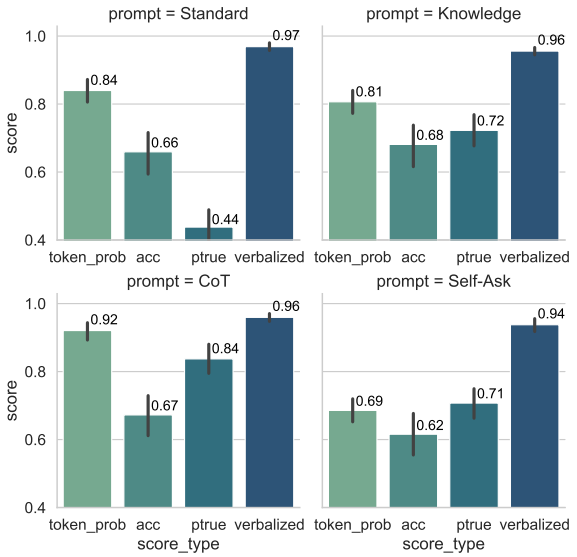

Figure 8: Accuracy and averaged confidence scores of different confidence extraction methods and prompting methods. In an ideal case with perfect calibration, the gap between these two scores should be zero.

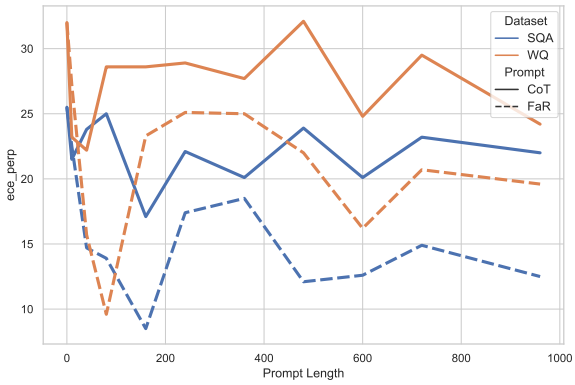

Figure 9: Expected Calibration Error (ECE) of CoT and FaR prompting with varying final prompt lengths on StrategyQA (SQA) and WQ.

address this, our proposed FaR prompting elicits the models to generate relevant internal knowledge and sources that serve as the proxy for such golden human-annotated facts.

A.5 Influence of Final-step Prompt Length

We identify the final prompt length as another confounder in our experiments: although different prompting styles are initialized with the same max length to output the thoughts, the thoughts injected in the final prompt asking for answers to the original questions may still have varied length.

To remove the potential influence of actual prompt length, we further compare CoT and FaR with varying prompt lengths of the final step, asking the model for the answers. Figure 9 presents the change of the ECE with respect to varying final-step prompt length. The table shows that FaR (dash lines) consistently outperforms CoT (solid lines) on different datasets and lengths. We can also observe that longer prompt length does not always lead to lower error, further validating our experimental results in Section 2.4 by removing the potential confounder: final prompt length.

A.6 Influence of the Number of Demonstrations

We identify the number of demonstrations in the prompt context as another important factor influencing the model calibration.

Method	Accuracy	ECE-Token.	ECE-P(True)	MacroCE-Token.	MacroCE-P(True)
Standard	60.3	25.5	35.1	67.8	41.5
Knowledge	69.9	27.8	38.1	78.9	68.9
Knowledge+Explain	65.1	18.9	35.4	**52.2**	76.8
CoT	67.2	25.0	34.3	82.6	42.0
Self-Ask	60.7	17.6	35.3	58.5	74.7
Self-Ask (aggregate)	60.4	17.0	35.0	60.0	72.0
Self-Con.	63.3	35.9	33.5	97.9	**37.2**
Pseudo-ToT	58.5	23.4	42.7	68.0	78.9
FaR(final)	64.0	**13.9**	**31.6**	52.9	41.0

Table 8: Expected Calibration Error (ECE) and Macro-average Calibration Error (MacroCE) of different prompting methods. The down arrow denotes the lower, the better. The best performance of each column is in **bold**. Token. and P(True) denote the use of Token Prob. and P(True) as the confidence extraction methods, respectively.

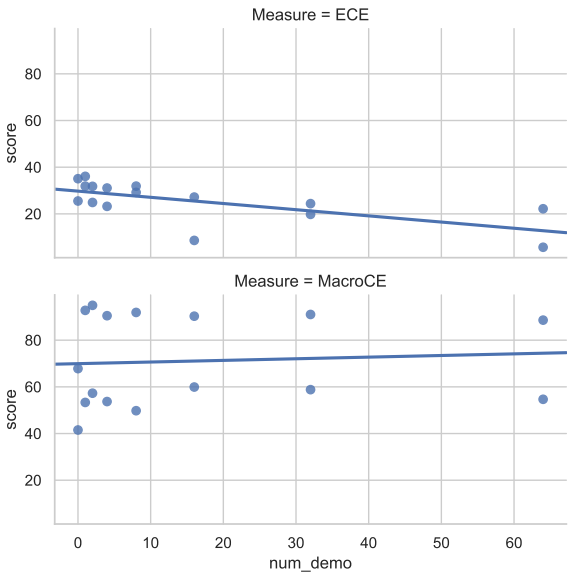

Figure 10: Expected Calibration Error (ECE) and Macro-average Calibration Error (MacroCE) of standard prompting with a varying number of demonstrations with Standard Prompting.

Figure 10 presents the change of the ECE and MacroCE with respect to the changing number of demonstrations with standard prompting. From the figure we can observe that: increasing the number of demonstrations helps reduce the error measured by ECE but not the error measured by MacroCE.

The reason can be that similar to the influence of the number of demonstrations on performance, demonstrations help the models do well on the questions that are answerable and show good consistency and robustness on these questions. The overall calibration is also improved with more robust answers. However, for extremely hard cases where the model can not solve at all, increasing the number of demonstrations can not help solve the task, but may still lead to increased confidence, hence the instance-level calibration is not improved.

A.7 Full Results of ECE and MacroCE of different prompting methods

In the main paper, we analyze the influence of prompting methods on confidence calibration from different perspectives, where for simplicity the results are presented by aggregating over different dimensions (e.g., averaging the metrics over different confidence extraction methods, or over different baselines). For completeness and for future reference, we report the full results of different prompting strategies (including our FaR) under different confidence extraction methods in Table 8, where we report both ECE and MacroCE.

In terms of confidence calibration, Table 8 shows that FaR (final) performs the best or the second best in all settings. On the other hand, in terms of model performance in accuracy, we also observe that FaR (final) outperforms 5 out of 8 baselines, but is slightly worse when compared to CoT or Knowledge-based prompting. The reason can be that (i) further constraints are applied to the model-generated thoughts and limit the degree of freedom in search; (ii) the model tends to provide conservative answers. **The slight loss in accuracy may be viewed as a kind of "calibration tax" for achieving better calibration.** Nevertheless, as discussed in Section 3.3, the conservative answers with expressed concerns can trigger and guide external knowledge augmentation on the (hard) examples with significant performance gain.

Question (Label)	Facts	Sources	Reflection	Output
Would Persephone be a good consultant to a landscape architect? (True)	1. Goddess of Spring and Vegetation: Persephone is often... 2. Queen of the underworld...	1. Homer's Hymn to Demeter: this ancient text... 2.Ovid's Metamorphoses: this collection of narratives...	1. Connection to Vegetation and Renewal: Persephone's role... 2. Cultural and Historical Influence:...	False. There will need to be further research.
Would an owl monkey enjoy a strawberry? (True)	1. Diet of Owl Monkeys: Owal monkeys primarily have a diet... 2. Taste Preferences of Primates...	1. Scientific journals and articles on primatology... (pubmed.ncbi.nlm.nih.gov) 2. Zoological Association websites...	1. Sweet Taste Preference in Primates... 2. Digestive Capabilities...	It is not possible to answer with current evidence this question.
Does Post Malone have a fear of needles? (False)	1. Tattoo Collection: he has a significant number of tattoos... 2. Public Statements:...	1. Interviews and Profiles: publications like Rolling Stone, Billboard, GQ... 2. Social Media: his accounts on platforms...	1. Nature of Tattoos: tattoos are created by repeatedly inserting a needle... 2. Lack of Public Statements regarding a fear of needles:...	False, but there is not yet sufficient evidence to answer.
Should a Celiac sufferer avoid spaghetti? (True)	1. Definition of Celiac Disease: Celiac disease is an autoimmune... 2. Gluten in Wheat and Gluten-free Alternatives:...	The recommendation on avoiding traditional spaghetti is supported by reputable sources like Mayo Clinic and NIDDK...	1. Nature of Celiac Disease:... Health Implications of Gluten Consumption: continuous consumption of gluten...	False (It depends on the ingredients of the spaghetti)

Table 9: Examples of the model output on the fact, reflection, and answer steps of FaR prompting. Detailed explanations on each point from the model are omitted for the presentation purpose.

A.8 Full Examples of FaR Prompting

In Figure 2 and Table 4, we demonstrate the general ideas about how FaR prompting works and how the model answers questions while expressing necessary concerns. Recall that FaR sequentially prompts the LLM for (i) facts, (ii) sources; (iii) reflection, and (iv) answers. In Table 9, we further provide the outputs at each of these steps for the examples in Table 4. We can observe that the output concern is relevant to the facts elicited in the context. For example, for the question, *Should a Celiac sufferer avoid spaghetti?*, the model says *it depends on the ingredients*, which relates to *Gluten-free alternatives for spaghetti* that is discussed in the fact step output of the model.