Think Twice Before Assure: Confidence Estimation for Large Language Models through Reflection on Multiple Answers

Anonymous ACL submission

Abstract

Confidence estimation aiming to evaluate output trustability is crucial for the application of large language models (LLM), especially the black-box ones. Existing confidence estimation of LLM is typically not calibrated due to the overconfidence of LLM on its generated incorrect answers. Existing approaches addressing the overconfidence issue are hindered by a significant limitation that they merely consider the confidence of one answer generated by LLM. To tackle this limitation, we propose a novel paradigm that thoroughly evaluates the trustability of multiple candidate answers to mitigate the overconfidence on incorrect answers. Building upon this paradigm, we introduce a two-step framework, which firstly instructs LLM to reflect and provide justifications for each answer, and then aggregates the justifications for comprehensive confidence estimation. This framework can be integrated with existing confidence estimation approaches for superior calibration. Experimental results on six datasets of three tasks demonstrate the rationality and effectiveness of the proposed framework.

1 Introduction

LLM suffers from the hallucination issue (Zhang et al., 2023c; Li et al., 2023a; Golovneva et al., 2022; Bang et al., 2023), which poses a significant challenge to the trustability of its outputs. A promising research direction for evaluating the output trustability is confidence estimation (Guo et al., 2017; Lin et al., 2022), which could be useful for identifying and rejecting unreliable outputs (Kamath et al., 2020). Given a question, confidence estimation aims to acquire LLM's confidence level on its generated answer, which reflects the LLM's certainty regarding the accuracy of the answer. The core of confidence estimation is to achieve calibration (Lin et al., 2022), ensuring that the confidence level aligns with the actual answer accuracy. In this paper, we aim to calibrate confidence estimation for black-box API LLMs due to their excellent performance (Achiam et al., 2023; OpenAI, 2024).

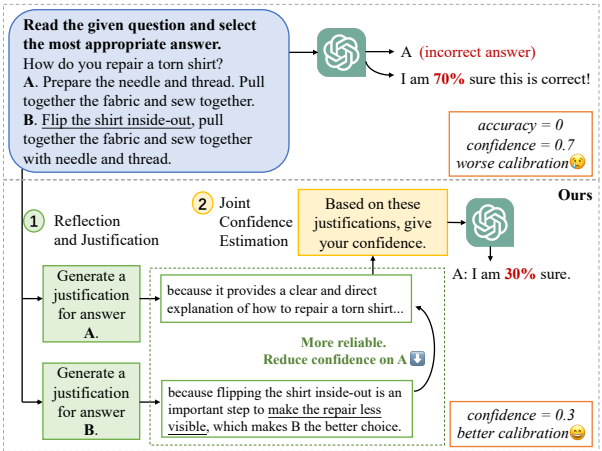

Figure 1: An illustration of our *Think Twice before Assure* framework for mitigating LLM overconfidence. LLM is instructed to reflect on the trustability of each answer before joint confidence estimation.

The key to achieving calibrated confidence estimation for black-box LLM lies in addressing the overconfidence issue. The LLM may be inherently biased towards trusting its generated answers (Mielke et al., 2022; Ling et al., 2023), making it hard to truly discern incorrect answers (Huang et al., 2023b) and exhibiting a tendency to assign overly high confidence scores to them (Si et al., 2022; Xiong et al., 2023). Previous studies attempting to tackle the overconfidence issue can be broadly categorized into two paradigms. The first paradigm mainly assumes that overconfidence is partly caused by the context bias between the prompt and the answer and thus performs prompt ensemble by constructing various instruction templates and diverse rephrasing of the question (Jiang et al., 2023; Zhao et al., 2023c). The second paradigm focuses on LLM self-evaluation, designing instructions such as asking LLM about the answer truthfulness (Kadavath et al., 2022) or examining the Chain-of-Thought (CoT) reasoning (Miao et al., 2023). However, both lines of research only

consider a single target answer generated by LLM, and the LLM may still contain bias towards the incorrect answers and be overconfident in them.

To tackle this limitation, we introduce a new multi-answer evaluation paradigm involving the consideration of multiple candidate answers to enhance confidence calibration (*cf.* Figure 2), where the evaluation of potentially correct answers helps to reduce the biased trust in the incorrect ones. This paradigm scrutinizes various answers on the trustability of being the correct response to the question, and aggregates these evaluations to derive a better confidence score for the target answer. The biased trust in the incorrect target answers can be alleviated through the trustability comparison with other more trustable answers. Our preliminary experiments reveal the efficacy of considering multiple answers to reduce overconfidence (*cf.* Section 2).

There are two key considerations in arriving at the proposed paradigm: resisting the inherent bias of LLM to precisely evaluate the trustability of each question-answer pair, and aggregating these assessments in the confidence estimation of the target answer. In this light, we present a novel confidence estimation framework to tackle the overconfidence issue of LLMs, named Think Twice before Assure (TTA) (*cf.* Figure 1). Our framework pushes LLM to reflect and justify from different answers' perspectives before confidence estimation on the target answer. Firstly, the LLM is instructed to generate justifications regarding the potential correctness of each answer. Subsequently, a prompt-based method is employed to integrate these justifications into joint confidence estimation for the target answer. Extensive experiments on six datasets across three tasks show improved calibration of TTA over methods from existing paradigms. Notably, TTA can be combined with other methods to further improve calibration. Our contributions are three-fold.

- We introduce a novel confidence estimation paradigm for mitigating the overconfidence issue in LLM, addressing the limitation of existing paradigms by reflection on multiple answers.
- We present a novel TTA framework to implement the multi-answer evaluation paradigm, which can be easily combined with existing methods.
- We conduct extensive experiments on three NLP tasks with six datasets, validating the rationality and effectiveness of the proposed framework.

2 Problem Formulation

Confidence Estimation for LLM. We formulate the task of confidence estimation for LLM as follows. Given the input comprising of question q combined with prompt p, which consists of an instruction and optional in-context examples, LLM can generate the answer a (Brown et al., 2020), denoted as the target answer. Thereafter, confidence estimation aims to obtain the LLM's confidence level on a, in the form of a confidence score $c \in \mathcal{R}$. Denoting the confidence estimation strategy as a function $CE(\cdot)$, this process can be abstracted as

$$a = LLM(p(q)), \tag{1}$$
$$c = CE(LLM(\cdot), p(q), a). \tag{2}$$

A common idea of $CE(\cdot)$ is to utilize the LLM output probability of a to estimate the confidence score (Kuhn et al., 2023; Hu et al., 2023), denoted as $c = Pr(LLM(\cdot), p(q), a)$. For black-box API LLM where the token probability is unavailable, this can be achieved by self-consistency (Wang et al., 2022; Si et al., 2022; Lin et al., 2023) and verbalized methods (Lin et al., 2022; Tian et al., 2023b). Self-consistency methods estimate the probability of answer a by sampling $D > 1$ responses from LLM (*e.g.,* using nucleus sampling (Holtzman et al., 2020)). Formally, we have

$$c = \frac{\sum_{i=1}^{D} \mathbb{1}(a_i = a)}{D}, \quad \text{where } a_i = LLM(p(q)). \tag{3}$$

Besides, the verbalized methods leverage a well-designed prompt p^b to instruct the LLM to output the K most likely answers and their corresponding probabilities in one response, *i.e.,*

$$[\{a_1, c_1\}, ...\{a_K, c_K\}] = LLM(p^b(q)). \tag{4}$$

where $[\cdot]$ denotes the concatenation of the K most likely answers with their probabilities.

Overconfidence Issue and Existing Solution Paradigms. However, LLMs are prone to be overconfident. Both self-consistency and verbalized methods have a severe overconfidence issue, where they exhibit high confidence in some incorrect answers (Si et al., 2022; Xiong et al., 2023). In fact, LLM has a bias to blindly trust its generated answers, leading to difficulties in distinguishing the answer correctness (Huang et al., 2023b; Ling et al., 2023; Mielke et al., 2022; Ren et al., 2023b).

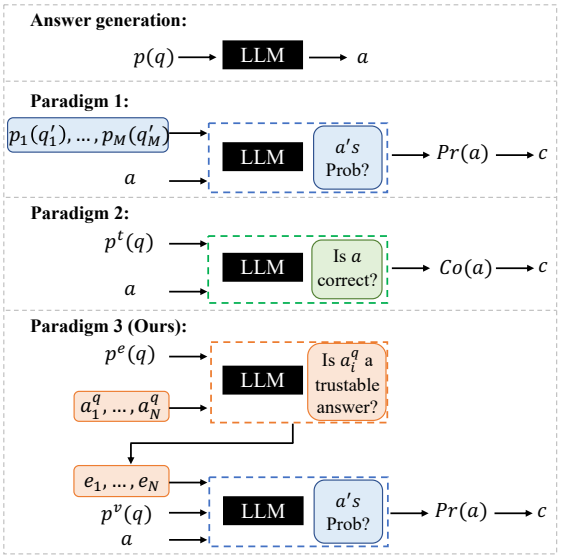

Figure 2: Two existing paradigms to tackle the overconfidence issue for LLM and our proposed multi-answer evaluation paradigm.

As a result, it causes a miscalibration between the confidence score and the answer accuracy.

We conclude the existing research addressing the overconfidence issue into two paradigms (*cf.* Figure 2). The first paradigm includes prompt ensemble methods. They posit that the overconfidence of LLM in a is influenced by the context bias between q and a (Zhao et al., 2023c) or p and a (Jiang et al., 2023). Therefore, they adopt different prompts, $\mathcal{P} = \{p_1, ..., p_M\}$, or various rephrasings of q, $\mathcal{Q}' = \{q'_1, ..., q'_M\}$, to alleviate the biased probability estimation of a. Assuming using M different inputs, the first paradigm estimates the confidence c by

$$c = \frac{1}{M}\sum_{i=1}^{M} Pr(LLM(\cdot), p_i(q'_i), a), \quad (5)$$

where $p_i \in \mathcal{P}, q'_i \in \mathcal{Q}'$.

The second paradigm involves self-evaluation, which utilizes instructions to guide LLM in self-evaluating the correctness of a from different perspectives, and uses the self-evaluated correctness as the confidence score, denoted as $Co(\cdot)$. Formally,

$$c = Co(LLM(\cdot), p^t(q), a). \quad (6)$$

where p^t denotes the evaluation prompt. This includes assessing the correctness of the CoT reasoning of a (Miao et al., 2023), completing masked questions using a (Weng et al., 2023), and checking input-output consistency (Manakul et al., 2023). To give an example, the P(True) method (Kadavath et al., 2022) asks LLM whether a is the true answer to q via the prompt p^r, and uses the probability of "True" in the sampled LLM responses as the confidence score, $c = Pr(LLM(\cdot), p^r(q, a), \textit{True})$. The two paradigms can also be combined for better calibration (Xiong et al., 2023; Chen and Mueller, 2023; Ren et al., 2023a; Agrawal et al., 2023).

A New Multi-Answer Evaluation Paradigm. A notable limitation of the existing two paradigms is that they merely focus on confidence estimation for a single LLM-generated answer a, in which LLM may be overconfident. Despite efforts in context bias elimination and self-evaluation, LLM's biased trust in the incorrect a may persist. However, we think that this biased trust could be alleviated if LLM had thoroughly compared the trustability of more candidate answers of q. If other answers had a strong tendency to be correct, the high confidence in a could be diminished, reducing the overconfidence risk. Therefore, we propose a novel multi-answer evaluation paradigm that considers N potential answers, denoted as $\{a_1^q, a_2^q, ..., a_N^q\}$ in confidence estimation[1]. First, LLM evaluates the trustability of each q, a_i^q pair using a designated prompt p^e. Then, all obtained evaluations $e_1, ..., e_N$ are aggregated to derive a more refined confidence score for a, using the prompt p^v.

$$c = Pr(LLM(\cdot), p^v(q, [e_1, ..., e_N]), a), \quad (7)$$
$$\text{where } e_i = LLM(p^e(q, a_i^q)), i \in \{1, ..., N\}.$$

This paradigm can also be combined with existing paradigms for better calibration (*cf.* Section 5.1).

Preliminary Experiments. We conduct a preliminary experiment to validate that considering more answers to adjust confidence scores is beneficial for calibration. Our hypothesis is that the confidence levels of other answers can be leveraged to identify and mitigate overconfidence in the incorrect a. To demonstrate this, we employ counterfactual questions with different labels. Counterfactual question $\bar{q}$ is minimally edited from q to have a different label with q. We aim to utilize the difference in q and $\bar{q}$'s labels to identify unreliable LLM answers and adjust the confidence. Suppose the LLM-generated answers for $\bar{q}$ and q are $\bar{a}$ and a, respectively. If $\bar{a}$ equals a, a and $\bar{a}$ must have at least one wrong

[1]In the case of multiple-choice questions, candidate answers are naturally provided. However, for questions without predefined choices, we can prompt the LLM to generate high-probability answers as candidates (Jiang et al., 2023).

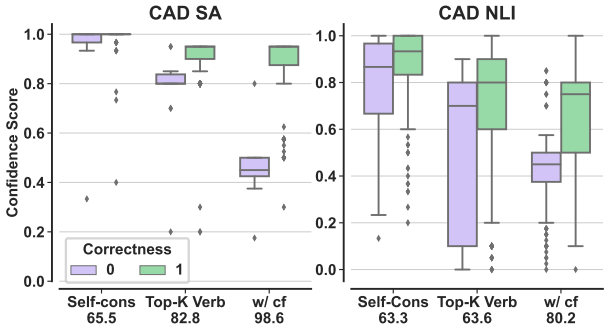

Figure 3: Comparison of confidence estimation methods on CAD. *w/ cf* denotes our strategy with counterfactual data. The AUROC is shown in the x-axis. The boxes on the left and right represent the confidence scores of incorrect and correct answers, respectively.

answer since $\bar{q}$ and q have different labels. Thus the confidence of a should be reduced according to the confidence of $\bar{a}$ because the increasing confidence of $\bar{a}$ indicates the weakened confidence of a. Conversely, if $\bar{a}$ differs from a, a and $\bar{a}$ are relatively trustable, and the confidence of a can be an average of itself and $\bar{a}$'s confidence. Denoting the confidences of a on $p(q)$ and $\bar{a}$ on $p(\bar{q})$ as c_a and $c_{\bar{a}}$, respectively, the confidence of a on $p(q)$ is re-calculated as

$$c = \begin{cases} \frac{1}{2}(c_a + c_{\bar{a}}) & \text{if } a \neq \bar{a}, \\ \frac{1}{2}(c_a + O(c_{\bar{a}})) & \text{else.} \end{cases} \quad (8)$$

where $O(c_{\bar{a}})$ denotes the confidence that $\bar{q}$'s label is not $\bar{a}$. In a k-classification task, we roughly estimate $O(c_{\bar{a}}) = \frac{1}{k-1}(1 - c_{\bar{a}})$.

We experiment with the CAD dataset (Kaushik et al., 2019), which contains human-annotated original and counterfactual data pairs for sentiment analysis (SA) and natural language inference (NLI) tasks. We compare the AUROC with self-consistency and Top-K verbalized methods to evaluate the confidence calibration of LLM (see Section 5 and Appendix B for more details). Figure 3 shows the performance and the statistics of confidence scores for correct and incorrect answers, from which we can observe that 1) the self-consistency and Top-K verbalized methods have notable overconfidence. The incorrect answers have similar confidence scores as correct answers, making it challenging to distinguish them. 2) Our strategy, denoted as *w/ cf*, improves AUROC by lowering confidence scores on incorrect answers, showing that considering more answers has the potential to alleviate the overconfidence issue in incorrect answers. However, human-annotated counterfactual data is not easily available, motivating us to propose the following framework.

3 Think Twice Before Assure Framework

Implementing the proposed paradigm involves two key considerations. First, given the potential bias of LLM being overconfident in the generated answer a, it is essential to develop strategies to resist this bias and thoroughly evaluate the trustability of each answer a_i^q. Secondly, it is crucial to derive strategies to effectively combine these evaluations for calibrated confidence estimation of a. To address these concerns, we introduce the following two-step framework.

Step 1: Reflection and Justification. We first instruct LLM to reflect on the trustability of each answer a_i^q and force LLM to seek justification for a_i^q as the correct answer of q, as defined by Eq. 7. The LLM is instructed with the prompt p^e in Table 1 to gather comprehensive evidence e_i from its knowledge, in order to support the rationality of using a_i^q to answer q. The rationality of this step is that p^e instructs LLM to abduct the justification from q and a_i^q, which avoids the LLM bias that lies in the generation direction from $p(q)$ to a. Generating CoT explanations from $p(q)$ before a has been validated to be ineffective for calibration (Zhang et al., 2020, 2023a).

p^e	The task is to [task description]. Question: [q]. Answer choices: [a_1^q, ..., a_N^q]. The answer is [a_i^q]. Please generate an explanation to try to justify the answer judgment.
p^v	The task is to [task description]. Provide your N best guesses and the probability that each is correct (0.0 to 1.0) for the following question... Question: [q]. Answer choices: [a_1^q, ..., a_N^q]. Possible explanation 1: [e^1] ... Possible explanation n: [e^N]

Table 1: Prompts used in our TTA framework. p^e prompts LLM to reflect and generate justification e_i for each a_i^q, and p^v prompts LLM to estimate confidence according to different e_i.

Step 2: Joint Confidence Estimation. After obtaining the justification e_i for each a_i^q, we proceed to integrate these e_i using the Top-K verbalized method (*cf.* Eq. 4) to derive the answer probability of a. We choose Top-K verbalized method due to its capability to generate a set of K potential answers along with their respective probabilities ef-

ficiently in a single response, where we set K as the number of answers N. As indicated in the prompt p^v of Table 1, the generated justifications e_i can be seamlessly integrated for confidence estimation.

An alternative approach to determine the final confidence score is to put one justification to each p^v, generating N distinct confidence scores for answer a, and then compute the average score.

$$c = \frac{1}{N}\sum_{i=1}^{N} Pr(LLM(\cdot), p^v(q, e_i), a) \quad (9)$$

In our experiments, we do not choose this setting, as prompting LLM to estimate from different perspectives via a unified prompt is more efficient and effective than a simple average of the confidence scores (further validated in Section 5.2). Moreover, we find that the confidence scores are sensitive to the order of justification in p^v, thus we shuffle the order of e^i in p^v and use the average confidence. Notably, the TTA framework can be combined with existing approaches, such as directly applying prompt ensemble, and Hybrid method which adjust the confidence based on the difference with other methods. (Xiong et al., 2023).

4 Related Work

Confidence Estimation of LLM. The idea of calibrated confidence estimation has been previously studies in neural networks (Guo et al., 2017) and applied in NLP models (Desai and Durrett, 2020; Dan and Roth, 2021; Hu et al., 2023). After the advent of LLM, many confidence estimation methods still utilize the output token probability, such as semantic uncertainty (Kuhn et al., 2023), temperature scaling (Shih et al., 2023), entropy-based method (Huang et al., 2023c), semantic significance (Duan et al., 2023), and fine-tuning based methods (Jiang et al., 2021; Lin et al., 2022). Our research is orthogonal to them, since we focuses on confidence estimation for black-box API LLM.

Others lines of research that are related but orthogonal to our approach include training independent models for LLM output evaluation (Wang and Li, 2023; Li et al., 2023b; Khalifa et al., 2023; Zhao et al., 2023b), and using external tools for LLM verification (Min et al., 2023; Ni et al., 2023). However, these works are usually applied to specific domains, while we aim at LLM self-calibration for general tasks. Also, there is research in fine-tuning the LLM for better trustability (An et al., 2023; Tian et al., 2023a), which is also orthogonal to us.

To tackle the overconfidence issue, the first category of methods also includes answer choice shuffling (Ren et al., 2023a), and reflection from multiple perspective (Zhang et al., 2024). Zhang et al. (2023b) also employ model ensemble for better calibration. The second category of method also includes program-like evaluation on CoT (Ling et al., 2023), generating and executing verification codes (Zhou et al., 2023), asking verification questions (Manakul et al., 2023), while some of them are limited to certain domains. Notably, the Top-K verbalized (Tian et al., 2023b), the self-consistency (Si et al., 2022), and their Hybrid (Xiong et al., 2023) methods also involve the confidence of other answers, yet the estimation of their confidences is also affected by the LLM bias and thus these answers do not genuinely contribute to the overconfidence mitigation of the target answer.

Application of LLM Confidence. Calibrated confidence score can be applied in many ways to avoid hallucination and erroneous outputs, such as identifying potentially hallucinated generation for knowledge retrieval and verification (Zhao et al., 2023a), guided output decoding (Xie et al., 2023), identifying ambiguous questions (Hou et al., 2023), selective generation (Ren et al., 2023a; Zablotskaia et al., 2023), and LLM self-improve (Huang et al., 2023a). More applications can be found in this survey (Pan et al., 2023).

5 Experiments

Setup. We conduct experiments on six datasets across three tasks. IMDB (Maas et al., 2011) and Flipkart (Vaghani and Thummar, 2023) for SA, SNLI (Bowman et al., 2015) and HANS (McCoy et al., 2019) for NLI, CommonsenseQA (Talmor et al., 2019) and PIQA (Bisk et al., 2020) for commonsense question answering (CQA). For LLMs, we utilize GPT-3.5 (*gpt-3.5-turbo-1106*), GPT-4 (*gpt-4-0613*) from OpenAI[2], and GLM-4 (Du et al., 2022) from ZhipuAI[3]. Dataset statistics and LLM parameters are listed in Appendices A.1 and A.2.

Compared Methods. We utilize the following categories of compared methods. Firstly, the baselines, including **Self-cons** (Wang et al., 2022) (*cf.* Eq. 3), **CoT-cons**, an extension of Self-cons by instructing LLM to output the CoT reasoning before the answer, **Top-K Verb** (Tian et al., 2023b) (*cf.*

[2] https://openai.com/blog/openai-api.
[3] https://open.bigmodel.cn/.

Eq. 4), and **Hybrid** (Xiong et al., 2023), an integration of Top-K Verb and Self-cons/CoT-cons, where we show the better results. Secondly, from the first paradigm, we have **Self-detect** (Zhao et al., 2023c), taking the answer entropy of multiple rephrased questions, and **CAPE** (Jiang et al., 2023), a prompt ensemble method that we implement on Top-K Verb. Thirdly, from the second paradigm, we have **P(True)** (Kadavath et al., 2022). We only compare with P(True) because most methods from the second paradigm are designed for specific domains or answers with CoT reasoning which are incompatible with our datasets. Finally, to show the flexibility of TTA in combining with existing methods to further improve calibration, we show the performance of Hybrid TTA with Top-K Verb (**TTA + Top-K Verb**), and TTA with prompt ensemble following CAPE (**TTA + PE**). For a fair comparison, we generate the target answer for each dataset with LLM temperature as 0, and compare all methods based on this target answer (*cf.* Eq 1). More details are provided in Appendices A.3 and A.4.

Evaluation Metrics. We use **AUROC** (Boyd et al., 2013) and **PRAUC** (Manning and Schutze, 1999) as evaluation metrics for confidence calibration, both ranging from 0 to 1. They assess the effectiveness of confidence scores in distinguishing answer correctness using true positive/false positive and precision/recall curves, respectively.

5.1 Results

Table 2 shows the performance of the compared methods on GPT-3.5. We can observe the followings. 1) TTA outperforms all compared methods in terms of both AUROC and PRAUC on IMDB and Flipkart for SA task, SNLI for NLI task, and CommonsenseQA for CQA task, demonstrating the effectiveness of TTA. 2) After combining TTA with other methods *i.e.,* Top-K Verb and PE, our method surpasses all compared methods on all datasets, showing the potential and flexibility of TTA in combining with others to further improving calibration. 3) Hybrid with Top-K Verb usually improves TTA's performance, which is in line with the performance improvement from Self-cons/CoT-cons to Hybrid. 4) CAPE is a very strong method, showing that the confidence estimation is largely influenced by the prompt. Combining TTA with PE usually improves TTA performance except for SNLI and Flipkart, which is in line with the performance decrease from Top-K Verb to CAPE. This is potentially related to the prompt sensitivity of these methods and the specific prompts adopted. 5) For other methods, CoT-cons outperforms Self-cons in 5 out of 6 datasets, as many tasks performs better with CoT reasoning. P(True) has ambivalent results which limits its applicability.

	IMDB		Flipkart	
	AUROC	PRAUC	AUROC	PRAUC
Self-cons	65.5	96.8	71.4	91.4
CoT-cons	75.6	97.7	72.8	91.9
Top-K Verb	82.8	98.5	79.3	93.7
P(True)	80.1	98.1	54.5	86.7
Hybrid	87.0	<u>98.8</u>	79.5	94.2
Self-detect	68.9	97.1	71.2	91.4
CAPE	87.7	**98.9**	76.4	93.9
TTA	87.9	**98.9**	<u>81.3</u>	<u>94.5</u>
TTA + Top-K Verb	<u>88.0</u>	**98.9**	**81.6**	**94.9**
TTA + PE	**88.1**	**98.9**	74.2	92.9

(a) SA.

	SNLI		HANS	
	AUROC	PRAUC	AUROC	PRAUC
Self-cons	63.3	71.4	56.0	64.8
CoT-cons	66.7	73.8	59.4	67.9
Top-K Verb	63.6	74.0	53.3	64.9
P(True)	55.4	67.4	60.8	70.1
Hybrid	66.7	78.8	62.0	71.1
Self-detect	59.3	68.5	55.3	64.5
CAPE	69.0	79.6	<u>71.9</u>	<u>80.1</u>
TTA	**77.9**	<u>84.6</u>	69.9	77.5
TTA + Top-K Verb	<u>77.1</u>	**84.7**	71.3	79.6
TTA + PE	70.8	76.7	**74.5**	**81.2**

(b) NLI.

	CommonsenseQA		PIQA	
	AUROC	PRAUC	AUROC	PRAUC
Self-cons	70.7	81.7	78.6	94.0
CoT-cons	81.8	88.9	76.7	94.2
Top-K Verb	69.4	81.5	76.8	93.3
P(True)	62.5	78.0	71.9	93.9
Hybrid	77.5	89.0	82.4	95.5
Self-detect	67.9	81.5	68.5	91.0
CAPE	78.7	88.8	<u>87.9</u>	<u>97.8</u>
TTA	83.5	90.7	83.4	95.2
TTA + Top-K Verb	**85.8**	**93.4**	85.3	96.2
TTA + PE	<u>84.4</u>	<u>92.1</u>	**90.3**	**97.9**

(c) CQA.

Table 2: Results of the compared methods on GPT-3.5. Bold font and underline indicate the best and second best performance, respectively.

5.2 In-depth Analysis

Ablation Studies. We conduct the following ablation studies to further validate the rationality of our framework design. 1) ***w/ CoT expl***: substituting $e_1, ..., e^N$ in p^v with N different CoT reasoning generated from $p(q)$ to reveal the rationality of re-

	IMDB		Flipkart	
	AUROC	PRAUC	AUROC	PRAUC
TTA	**87.9**	**98.9**	**81.3**	**94.5**
w/ CoT expl	72.4	97.5	76.6	93.4
sep expl	86.5	98.8	79.5	94.2
w/o shuffle	75.9	98.3	71.7	92.0

(a) SA.

	SNLI		HANS	
	AUROC	PRAUC	AUROC	PRAUC
TTA	**77.9**	**84.6**	**69.9**	**77.5**
w/ CoT expl	67.1	75.2	53.7	64.1
sep expl	68.5	75.3	54.1	63.8
w/o shuffle	70.6	77.6	60.7	67.9

(b) NLI.

	CommonsenseQA		PIQA	
	AUROC	PRAUC	AUROC	PRAUC
TTA	**83.9**	90.9	83.4	95.2
w/ CoT expl	78.7	86.8	81.3	94.8
sep expl	83.3	**92.0**	**84.0**	**95.8**
w/o shuffle	80.3	87.8	80.4	94.3

(c) CQA.

Table 3: Ablation studies.

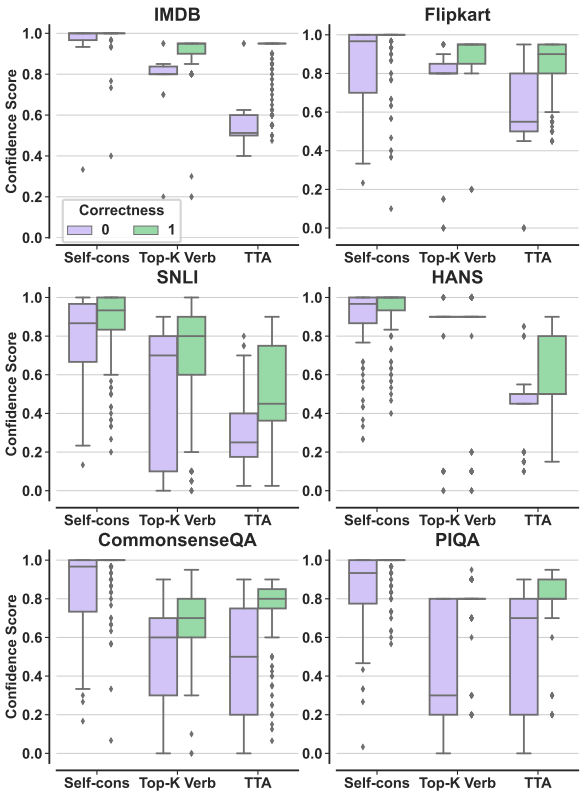

Figure 4: Visualization of bias mitigation effect of TTA which largely reduces the confidence overlaps between correct (right) and incorrect (left) answers.

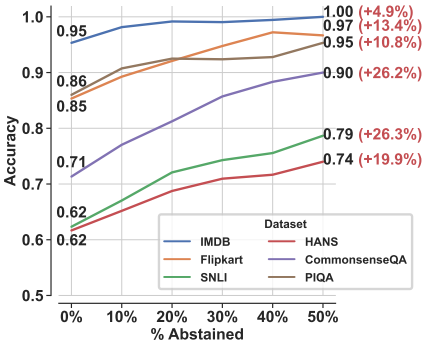

Figure 5: Accuracy improvement of selective prediction on TTA confidence scores.

flection on various answers. 2) ***sep expl***: placing a single e_i in p^v each time and calculating the averaged confidence score to reveal the effectiveness of joint considering all e_i in one p^v. 3) ***w/o shuffle***: ablating the order shuffling of e_i in p^v.

From Table 3, we can observe that: 1) *w/ CoT expl* largely underperforms TTA on all three tasks, demonstrating the rationality of pushing LLM to reflect and justify from each answer's perspective. 2) *sep expl* underperforms TTA on both SA and NLI tasks, showing that jointly considering multiple justifications in one prompt is often more beneficial, and thus we choose this setting. It slightly outperforms TTA on the CQA task, potentially due to the higher independency and objectivity of the answer choices. 3) *w/o shuffle* also underperforms TTA, indicating that there exists order sensitivity for e_i, and shuffling their order improves calibration by mitigating their position bias.

Effect on Bias Mitigation. Since our goal of mitigating the overconfidence issue is to reduce the extremely high confidence scores on incorrect answers, we show the statistics of the confidence scores for each dataset regarding the answer correctness in Figure 4 to reveal the mechanism of TTA. We compare TTA with Self-cons and Top-K Verb which are witnessed with overconfidence. We can observe that TTA clearly reduces the confidence overlaps between correct and incorrect answers on all datasets, and significantly decreases the confidence scores on incorrect answers in IMDB, Flipkart, SNLI and HANS. Thus, the answer accuracy is more separable by the confidence score, achieving better calibration.

Effect on Selective Prediction via Confidence Score. To show the utility of the confidence score, we conduct experiments in selective prediction. The idea of selective prediction is to refrain from adopting the answers from LLM with low confidence to maintain better accuracy of the remaining answers. In Figure 5, we show the accuracy of the remaining answers by abstaining 0% - 50% of an-

		Flipkart	HANS	CommonsenseQA
a^{sc}	Self-cons	72.7	52.7	68.2
	CoT-cons	74.4	57.5	80.4
	Top-K Verb	80.4	51.8	69.2
	TTA	**82.2**	**69.5**	**82.7**
a^{cc}	Self-cons	78.3	57.0	68.1
	CoT-cons	79.2	57.8	74.3
	Top-K Verb	83.9	53.3	67.5
	TTA	**84.3**	**69.2**	**75.0**

Table 4: AUROC on two different target answers.

swers with the lowest confidences from TTA. We can observe that by increasing the percentage of abstained answers, the accuracy for these datasets gradually improves around 10% - 30%, and IMDB even achieves 100% accuracy. Naturally, the increase for datasets with lower accuracy is generally easier than datasets with higher accuracy. The result shows that TTA possess strong potential to be applied in selective prediction scenarios.

Analysis on the Robustness of TTA. We evaluate the robustness of TTA from three aspects: different target answers, different LLMs, and parameter sensitivity. In addition, we examine prompt sensitivity in Appendix C.

Firstly, the generation of target answer a may vary under LLM randomness, *e.g.,* setting the temperature greater than 0. We verify the robustness of TTA by utilizing **different target answers**, *i.e.,* the majority answer of Self-cons (a^{sc}) and CoT-cons (a^{cc}), respectively, as shown in Table 4. We can observe the following. 1) For both sets of target answers, TTA largely outperforms baselines, showing its effectiveness. 2) Different target answers may have very different calibration performance. Specifically, a^{cc} on CommonsenseQA has a sharp decrease in AUROC of TTA and CoT-cons compared with the other target answers, which is probably due to the majority voting with CoT explanation diminished the the effect of reflection and justification in calibration.

Secondly, we evaluate TTA on **different LLMs**, *i.e.,* GPT-4 and GLM-4. Table 5 shows the performance of Flipkart on its top two compared methods. We can observe that across different LLMs, TTA outperforms baseline methods, and further hybrid with Top-K Verb outperforms the Hybrid method, validating its effectiveness. Moreover, Hybrid does not stably outperform single method across LLMs.

Thirdly, we evaluate the **parameter sensitivity** of TTA by changing the number of justifications and number of guesses in p^v. We conduct

	GPT-4		GLM-4	
	AUROC	PRAUC	AUROC	PRAUC
Top-K Verb	80.8	94.3	81.1	92.1
Hybrid	81.7	94.7	80.4	92.0
TTA	81.0	94.5	**83.3**	**93.4**
TTA + Top-K Verb	**82.2**	**94.9**	82.7	93.2

Table 5: Performance comparison of Flipkart on different LLMs.

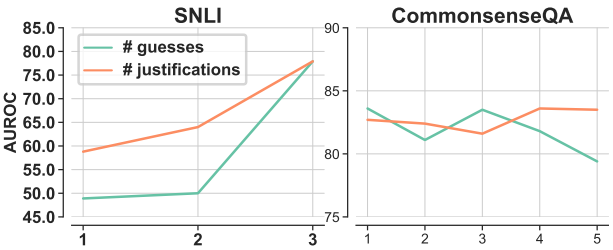

Figure 6: Parameter sensitivity, *i.e.,* changing the number of justifications and number of guesses in p^v.

experiments on CommonsenseQA with five answer choices, and SNLI with three answer choices. From Figure 6, we can observe the followings. 1) A larger number of justifications increases the model performance on both SNLI and CommonsenseQA datasets, indicating a sufficient number of justifications is vital for better calibration. 2) Increasing the number of guesses results in a significant performance improvement on the SNLI dataset, revealing that enough number of guesses is demanded for the NLI task. 3) Comparably, the change in the number of guesses has a slight effect on the performance of the CommonsenseQA dataset, which is potentially because the CQA task is more objective than NLI.

6 Conclusion

In this paper, we tackled the overconfidence issue of confidence estimation on black-box API LLMs. We categorized existing methods into two paradigms and pointed out their limitation of merely estimating for a single target answer with potential LLM overconfidence. We proposed a novel paradigm to address this limitation by evaluating the trustability of multiple candidate answers. Following our paradigm, we presented a two-step framework TTA by asking LLM to reflect and justify the trustability of each answer for joint confidence estimation. Our framework achieved improved calibration performance over compared methods and was combined with existing methods for further improvement. In future work, we will explore the combination of TTA with more methods, and its utility in white-box LLMs.

Limitations

Our work has several limitations. Firstly, our research scope is limited to the confidence estimation for black-box API LLM. While our framework is suitable for many state-of-the-art LLMs in this form, it might not be optimal for white-box LLMs, which offer access to token probabilities, thus limiting its broader applicability. Secondly, the utility of confidence estimation is not primarily studies in this work. Although we demonstrate the utility of confidence scores in selective prediction scenarios, the challenge still lies in leveraging them to enhance task accuracy or enable LLM self-correction, calling for further exploration. Lastly, our framework lacks consideration in prompt optimization for calibration, an area where future confidence estimation methods are supposed to consider.

Ethics Statement

Our ethical concerns involve the following. First, our experimental results are mainly obtained in English datasets, where the applicability on other languages are not comprehensively evaluated. Secondly, our research scope is black-box API LLMs, where open-sourced LLMs are more advocated for its reproducibility. Finally, the confidence estimation of LLM may mislead people to blindly trust LLM and easily accept untrustable answers, causing potential harms.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ayush Agrawal, Lester Mackey, and Adam Tauman Kalai. 2023. Do language models know when they're hallucinating references? *arXiv preprint arXiv:2305.18248*.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. *arXiv preprint arXiv:2310.20689*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Kendrick Boyd, Kevin H Eng, and C David Page. 2013. Area under the precision-recall curve: point estimates and confidence intervals. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 451–466. Springer.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jiuhai Chen and Jonas Mueller. 2023. Quantifying uncertainty in answers from any language model via intrinsic and extrinsic confidence assessment. *arXiv preprint arXiv:2308.16175*.

Soham Dan and Dan Roth. 2021. On the effects of transformer size on in-and out-of-domain calibration. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2096–2101.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. Decomposing uncertainty for large language models through input clarification ensembling. *arXiv preprint arXiv:2311.08718*.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications. *arXiv preprint arXiv:2306.04459*.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023a. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023b. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Yuheng Huang, Jiayang Song, Zhijie Wang, Huaming Chen, and Lei Ma. 2023c. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Baker Grosse, and Jimmy Ba. 2023. Calibrating language models via augmented prompt ensembles.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 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.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Grace: Discriminator-guided chain-of-thought reasoning.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of statistical natural language processing*. MIT press.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2023. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-tau Yih, Sida Wang, and Xi Victoria Lin. 2023. Lever: Learning to verify language-to-code generation with execution. In *International Conference on Machine Learning*, pages 26106–26128. PMLR.

OpenAI. 2024. Chatgpt.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *CoRR*, abs/2308.03188.

Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. 2023a. Self-evaluation improves selective generation in large language models. *arXiv preprint arXiv:2312.09300*.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023b. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

Andy Shih, Dorsa Sadigh, and Stefano Ermon. 2023. Long horizon temperature scaling. *arXiv preprint arXiv:2302.03686*.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. In *The Eleventh International Conference on Learning Representations*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher Manning, and Chelsea Finn. 2023a. Fine-tuning language models for factuality. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

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

Nirali Vaghani and Mansi Thummar. 2023. BFlipkart product reviews with sentiment dataset. https://www.kaggle.com/dsv/4940809.

Danqing Wang and Lei Li. 2023. Learning from mistakes via cooperative study assistant for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10667–10685.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Polina Zablotskaia, Du Phan, Joshua Maynez, Shashi Narayan, Jie Ren, and Jeremiah Liu. 2023. On uncertainty calibration and selective generation in probabilistic neural summarization: A benchmark study. *arXiv preprint arXiv:2304.08653*.

Hanlin Zhang, Yi-Fan Zhang, Yaodong Yu, Dhruv Madeka, Dean Foster, Eric Xing, Hima Lakkaraju, and Sham Kakade. 2023a. A study on the calibration of in-context learning. *arXiv preprint arXiv:2312.04021*.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A. Malin, and Sricharan Kumar. 2023b. SAC^3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024. Self-contrast: Better reflection through

inconsistent solving perspectives. *arXiv preprint arXiv:2401.02009*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023c. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 295–305.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023a. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Theodore Zhao, Mu Wei, J Samuel Preston, and Hoifung Poon. 2023b. Automatic calibration and error correction for large language models via pareto optimal self-supervision. *arXiv preprint arXiv:2306.16564*.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2023c. Knowing what llms do not know: A simple yet effective self-detection method. *arXiv preprint arXiv:2310.17918*.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, et al. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *arXiv preprint arXiv:2308.07921*.

A Details for compared methods.

A.1 LLM Parameters.

For all LLMs, we set the maximum token as 200. For GPT-3.5 and GPT-4, if sampling a single response ($N = 1$), we set the temperature as 0, and other parameters as default. If sampling multiple responses, we sample $N = 30$ responses with temperature as 1, which is only for Self-cons, CoT-cons, P(True). Specially, for Self-detect we sample 15 rephrasing for each question with temperature as 1, and one answer for each rephrased question with temperature as 0, following the original paper. For GLM-4, if sampling a single response, we set the do_sample as False. If sampling a variety of responses, we set temperature as 0.9 and top p as 0.9.

	N	examples
IMDB	2	positive negative
Flipkart	2	positive negative
SNLI	3	entailment, neutral, contradiction
HANS	2	entailment, non entailment
CommonsenseQA	5	(a) yard, (b) basement, (c) kitchen, (d) living room, (e) garden
PIQA	2	(a) pour it onto a plate, (b) pour it into a jar

Table 6: The number (N) and examples of candidate answers for each dataset.

Note that these LLM parameters are not carefully tuned.

A.2 Dataset Detail.

Due to the cost limitation, we randomly sample 300 training data for each dataset in our experiments. For IMDB and SNLI datasets, we use the same randomly sampled 300 data sets as the CAD SA and NLI in the preliminary experiments. We will release the dataset splits. Table 6 shows the number and examples of candidate answers for each dataset.

A.3 Prompts

The basic instructions for different datasets are shown as below, where [] refers to specific task inputs.

- **IMDB**:
  Given a piece of movie review, classify the attitude to the movie as Positive or Negative. [text]

- **Flipkart**:
  Given a piece of text, classify the sentiment as Positive or Negative. [text]

- **SNLI**:
  Determine whether the hypothesis is an entailment (can be logically inferred from the premise), a contradiction (cannot be true given the premise), or neutral (does not have enough information to determine its truth value). Premise: [premise] Hypothesis: [hypopthesis].

- **HANS**:
  Determine whether the second sentence in each pair logically follows from the first sentence. The output is either "entailment" if the second sentence logically follows from the first, or "not entailment" if it does not.

Sentence 1: [sentence1]. Sentence 2: [sentence2].

- **CommonsenseQA**:
  Read the given question and select the most appropriate answer by indicating the associated letter. Question: [question]. Answer choices: (a) a_1^q (b) a_2^q (c) a_3^q (d) a_4^q (e) a_5^q.

- **PIQA**:
  Read the given question and select the most appropriate answer by indicating the associated letter. Question: [question]. Answer choices: (a) a_1^q (b) a_2^q.

The prompts for compared methods are shown below, where *[instruction]* denotes the task instruction with the task input, and *[instruction_only]* denotes the instruction without task input.

- **Self-cons**: *[instruction].*

- **CoT-cons**:
  [instruction]. Please output strictly following this format: Explanation: [reasons for the sentiment label] Answer: [Positive or Negative]

- **Top-K Verb** p^b:
  The task is to [instruction_only]. Provide your n best guesses and the probability that each is correct (0.0 to 1.0) for the following question. Give ONLY the guesses and probabilities, no other words or explanation. For example:
  G1: <first most likely guess, as short as possible; not a complete sentence, just the guess!>
  P1: <the probability between 0.0 and 1.0 that G1 is correct, without any extra commentary whatsoever; just the probability!> ... GN: <N-th most likely guess, as short as possible; not a complete sentence, just the guess!>
  PN: <the probability between 0.0 and 1.0 that GN is correct, without any extra commentary whatsoever; just the probability!> [question] [answer choices].

- **P(True)** p^t:
  The task is to [instruction]. Label: [label]. Is the label correct or incorrect?

- **Self-detect**:
  For question rephrasing: *Paraphrase the given sentence. Please make sure the paraphrased sentence has exactly the same meaning as the original sentence. [question]*
  For inference: *[instruction].*

- **CAPE**:
  Provide your 2 best guesses and the probability that each is correct (0.0 to 1.0) for the following task. Give ONLY the guesses and probabilities, no other words or explanation. For example:
  G1: <first most likely guess, as short as possible; not a complete sentence, just the guess!>
  P1: <the probability between 0.0 and 1.0 that G1 is correct, without any extra commentary whatsoever; just the probability!> ... GN: <N-th most likely guess, as short as possible; not a complete sentence, just the guess!>
  PN: <the probability between 0.0 and 1.0 that GN is correct, without any extra commentary whatsoever; just the probability!> Instruction: [instruction_only] [question]
  - a_1^q (or A. a_1^q)
  ...
  - a_N^q (or N. a_N^q)
  Possible explanation 1: [e^1]
  ...
  Possible explanation N: [e^N]
  Correct Choice:

- **TTA** p^v:
  The task is to [instruction_only]. Provide your n best guesses and the probability that each is correct (0.0 to 1.0) for the following question. Give ONLY the guesses and probabilities, no other words or explanation. For example:
  G1: <first most likely guess, as short as possible; not a complete sentence, just the guess!>
  P1: <the probability between 0.0 and 1.0 that G1 is correct, without any extra commentary whatsoever; just the probability!> ... GN: <N-th most likely guess, as short as possible; not a complete sentence, just the guess!>
  PN: <the probability between 0.0 and 1.0 that GN is correct, without any extra commentary whatsoever; just the probability!>
  [question] [answer choices].
  Possible explanation 1: [explanation 1].
  ...
  Possible explanation N: [explanation N].

A.4 Additional Implementation Detail.

For TTA and Top-K Verb, the N is set to the number of candidate answers for each dataset as in Table 6.

For the shuffling of the justification order in p^v, we use one original and one reversed order for TTA on all datasets. For datasets with more than two justifications (SNLI and CommonsenseQA), we set the original justification order for SNLI as "entailment, neutral, contradiction" and follow the given answer choice order for CommonsenseQA in the dataset.

CAPE is prompt ensemble for Top-K Verb. We follow the original paper to adopt two multi-choice template with alphabetic or itemized labels in addition to the original Top-K Verb prompt (See Section A.3). For each multi-choice template, we use the original and the reversed label orders. In total, the confidence score is an average of five prompts.

For TTA + PE, we put TTA into the multi-choice template with alphabetic labels, and use two reversed label orders and 2 reversed justification orders, in total four prompts.

The number of API calls for different methods are shown in Table 7.

	Self-cons	CoT-cons	Top-K Verb	P(True)	Hybrid
# call	30	30	1	30	31
	Self-detect	CAPE	TTA	TTA + Top-K Verb	TTA + PE
# call	30	5	N+2	N+3	N+4

Table 7: Comparison on the number of API calls of compared methods, where n denotes the number of choices for different datasets.

B Implementation Detail for Preliminary Experiments.

For the preliminary experiments, we randomly sample 300 instances from the training set of CAD SA and NLI, respectively. For those original questions with more than one counterfactual questions, we randomly select one counterfactual question for experiment. The prompts can be viewed in Section A.3. CAD SA is annotated from IMDB, and CAD NLI is annotated from SNLI. The *w/ cf* is based on Top-K Verb, which is better calibrated than Self-cons. For *w/ cf*, we obtain the Top-K Verb outputs for counterfactual and original questions, respectively. We use the guess with the largest probability in the response as the answer to the question (a for q and $\bar{a}$ for $\bar{q}$), and the probability as its confidence score. The LLM is GPT-3.5 (*gpt-3.5-1106*). See Section A.1 for LLM parameters.

	PIQA	HANS	Flipkart
p^e	84.2 ± 2.0	62.7 ± 4.3	78.0 ± 2.2
p^v	83.0 ± 0.5	68.3 ± 1.7	81.2 ± 0.3

Table 8: The average and standard deviation of AUROC for TTA with different rephrasing of prompts.

C Prompt Sensitivity

We examine the prompt sensitivity of p^e and p^v by rephrasing each of them three times with ChatGPT[4] and compute the average and standard deviation of AUROC, as shown in Table 8. We can observe the followings. 1) The variation of prompts has a mild effect on the calibration of TTA. Across the three datasets, HANS is the most sensitive to prompt rephrasing, potentially related to its lower calibration performance. 2) The change of p^e has larger impact on calibration than p^v. This is probably because the justifications generated by p^e have a larger space of variation than the outputs of p^v, *i.e.,* guesses and probabilities.

D Case study

We present two case study of PIQA. From Table 9, we can observe that the confidence of the incorrect answer (a) is lowered by TTA ($0.7 \rightarrow 0.45$). The justification (b) points out the reason why (b) is the preferred answer, which increased the confidence in (b) and in turn decreases (a)'s confidence. From Table 10, we can observe that the LLM is not sure whether the sponge should be dampened, thus having an ambivalent confidence score of 0.5 is better than being 0.7 confident on (b).

[4] https://chat.openai.com/.

Input: The task is to read the given question and select the most appropriate answer by indicating the associated letter. Provide your 2 best guesses and the probability that each is correct (0.0 to 1.0) for the following question. Give ONLY the guesses and probabilities, no other words or explanation. For example:
G1: <first most likely guess, as short as possible; not a complete sentence, just the guess!>
P1: <the probability between 0.0 and 1.0 that G1 is correct, without any extra commentary whatsoever; just the probability!> ... G2: <2-th most likely guess, as short as possible; not a complete sentence, just the guess!>
P2: <the probability between 0.0 and 1.0 that G2 is correct, without any extra commentary whatsoever; just the probability!>
Question: How do you repair a torn shirt?
Answer Choices:
(a) To repair a torn shirt, prepare the needle with a piece of thread tied to it. Pull together the fabric where the shirt is torn and sew together.
(b) Prepare the needle with a piece of thread tied to it. Flip the shirt inside-out, pull together the fabric where the shirt is torn and sew together.
Prediction: (a). Ground truth: (b).

Top-K Verb Output: G1: A P1: 0.7 G2: B P2: 0.3.
Confidence: 0.7

Justification for (a): The most appropriate answer is (a) because it provides a clear and direct explanation of how to repair a torn shirt. It instructs to prepare the needle with a piece of thread tied to it and then pull together the fabric where the shirt is torn and sew it together. This answer is straightforward and does not include unnecessary steps, making it the most appropriate choice.
p^v **output 1**: G1: A P1: 0.6 G2: B P2: 0.4
Justification for (b): Option (b) is the most appropriate answer because it provides a more detailed and accurate description of the process of repairing a torn shirt. It includes the step of flipping the shirt inside-out, which is an important step in order to hide the stitches and make the repair less visible. This additional detail makes option (b) the better choice for repairing a torn shirt.
p^v **output 2**: G1: B P1: 0.7 G2: A P2: 0.3
TTA Confidence: (0.6 + 0.3)/2 = 0.45

Table 9: Case study for PIQA. p^v output 1 refers to p^v with explanation (a) before explanation (b), and p^v output 2 refers to the reversed order.

Input: The task is to read the given question and select the most appropriate answer by indicating the associated letter. Provide your 2 best guesses and the probability that each is correct (0.0 to 1.0) for the following question. Give ONLY the guesses and probabilities, no other words or explanation. For example:
G1: <first most likely guess, as short as possible; not a complete sentence, just the guess!>
P1: <the probability between 0.0 and 1.0 that G1 is correct, without any extra commentary whatsoever; just the probability!> ... G2: <2-th most likely guess, as short as possible; not a complete sentence, just the guess!>
P2: <the probability between 0.0 and 1.0 that G2 is correct, without any extra commentary whatsoever; just the probability!>
Question: How do I keep my favorite couch fur free?
Answer Choices:
(a) Purchase a dry sponge at a pet supply store and use it to lightly wipe dog hair in a sweeping motion from upholstery.
(b) Purchase a dry sponge at a pet supply store and dampen it and use it to lightly wipe dog hair in a sweeping motion from upholstery.
Prediction: (b). Ground truth: (a).

Top-K Verb Output: G1: B P1: 0.7 G2: A P2: 0.3.
Confidence: 0.7

Justification for (a): The most appropriate answer is (a) because using a dry sponge to wipe dog hair in a sweeping motion from upholstery will help to remove the fur without getting the couch wet. Dampening the sponge, as suggested in option (b), may lead to the couch getting wet and potentially causing damage to the upholstery. Therefore, option (a) is the best choice for keeping the favorite couch fur free.
p^v **output 1**: G1: A P1: 0.7 G2: B P2: 0.3
Justification for (b): The most appropriate answer is (b) because dampening the dry sponge will help to better pick up and remove the dog hair from the upholstery. Using a dry sponge alone may not be as effective in removing the fur. Therefore, dampening the sponge will provide better results in keeping the favorite couch fur free.
p^v **output 2**: G1: B P1: 0.7 G2: A P2: 0.3
TTA Confidence: (0.3 + 0.7)/2 = 0.5

Table 10: Case study for PIQA. p^v output 1 refers to p^v with justification (a) before justification (b), and p^v output 2 refers to the reversed order.