Towards Fully Exploiting LLM Internal States for Enhanced Knowledge Boundary Perception

Anonymous ACL submission

Abstract

The internal states of large language models (LLMs) have been shown to effectively indicate the factuality of their responses. However, current research on LLMs' perception of their knowledge boundaries through hidden states has two main limitations: 1) It lacks guidance on the timing for extracting these hidden states—whether before or after response generation—and 2) It has not introduced methods for further enhancing this perception. To address these gaps, we: 1) Analyze LLMs' ability to assess the factuality of their answers based on internal representations of the question alone versus the entire question-answer sequence. Our experiments on three factual QA benchmarks show that LLMs can predict the correctness of their responses before generation, offering a cost-efficient approach by avoiding inference, with this ability further refined after generation. 2) Propose C^3 (Consistency-based Confidence Calibration), a technique that adjusts the model's confidence by measuring the consistency of its confidence across different reformulations of the same question. Experimental results demonstrate that C^3 consistently improves LLMs' self-assessment accuracy. We recommend using pre-generation predictions in high-efficiency scenarios and applying C^3 in safety-critical applications.

1 Introduction

Recently, Large Language Models (LLMs) have demonstrated remarkable performance across a range of NLP tasks (Touvron et al., 2023; Achiam et al., 2023; Yang et al., 2024), sometimes even surpassing human capabilities. However, unlike humans, LLMs do not have a clear understanding of their knowledge boundaries (i.e., knowing what they know and what they do not know). A reliable system should provide correct answers to the questions it knows and refuse to answer the questions it does not know. This requires the model to align its confidence with its actual abilities.

Current research on a model's perception of its knowledge boundaries mainly involves two types of confidence: probabilistic confidence where they use the probability of generated tokens as the model's confidence in the answer (Guo et al., 2017; Desai and Durrett, 2020; Jiang et al., 2021; Kadavath et al., 2022; Si et al., 2022; Kuhn et al., 2023) and verbalized confidence where they teach the model to express its confidence in the answer using natural language. (Lin et al., 2022; Yin et al., 2023; Tian et al., 2023; Xiong et al., 2023; Yang et al., 2023; Ni et al., 2024a)

Recently, some studies have shown that LLMs' internal states contain their judgment regarding the factuality of the text (Azaria and Mitchell, 2023; Chen et al., 2024; Wang et al., 2024). Su et al. (2024) extended this approach to detect the factuality of model's self-generated text and found that, compared to previous signals, LLMs' internal states more accurately reflect their confidence in the answer. Therefore, in this paper, we investigate LLMs' perception of their knowledge boundaries based on the internal states.

Current research on LLMs' perception of their knowledge boundaries through hidden states has two main limitations: 1) It lacks guidance on the timing for extracting these hidden states—whether before or after response generation. If it is possible to determine that the model cannot provide a correct answer before it responds, generation can be halted to save computational cost and prevent the generation of incorrect information. 2) It has not introduced methods for further enhancing this perception. Therefore, we try to tackle these two problems in this paper.

To investigate the timing for extracting hidden states, we estimate the LLMs' perception in providing a correct answer based on the model's semantic representation of the question alone (i.e., pre-generation perception) and the entire question-response sequence (i.e., post-generation percep-

tion). We conduct experiments on Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (HQ) (Yang et al., 2018), and MMLU (Hendrycks et al., 2020) to examine the effects of different question difficulties and task formats. Additionally, we employ Chain-of-Thought (Wei et al., 2022) strategy to increase the information content in generated responses, aiming to explore whether this added information impacts the gap between these two perceptions. Experimental results reveal that LLMs exhibit a good level of pre-generation perception and their post-generation perception will be further enhanced. The gap between pre- and post-generation perception is not significantly affected by question difficulty, task format, or the amount of information in the generated content. Therefore, in efficiency-critical scenarios, pre-perception can be used to determine whether generation is necessary. Compared to post-generation assessment, this approach is more efficient, especially when generating lengthy content.

In safety-critical domains, avoiding as many potential risks as possible is more important than improving efficiency. Therefore, we introduce C^3 (Consistency-based Confidence Calibration) to enhance LLMs' perception level. The method hinges on confidence consistency: if a model is confident in its initial answer but loses confidence when the question format changes, this inconsistency signals potential uncertainty, suggesting the initial judgment may be wrong.

To apply C^3, we prompt the model to generate multiple possible answers for a given question, which are then used to form a set of options for a multiple-choice version of the same question. By varying the number of options, we create different question formats to gather supplementary information. We then aggregate the model's confidence across these multiple-choice versions to calibrate its confidence in the original open-ended answer. Experiments on NQ and HotpotQA show that C^3 effectively reduces overconfidence, improves the model's perception of its knowledge boundaries, and significantly reduces potential risks. We recommend using pre-generation predictions in high-efficiency scenarios and applying C^3 in safety-critical applications.

2 Related Work

The most important aspect of evaluating and enhancing a model's perception of its knowledge boundaries is how to accurately extract and calibrate the model's confidence in its ability to provide the correct answer. Existing research can be primarily categorized into the following three types:

Probabilistic Confidence. This series of work uses the generation probability of the answer as the model's confidence. (Guo et al., 2017; Desai and Durrett, 2020; Jiang et al., 2021; Kadavath et al., 2022; Si et al., 2022; Kuhn et al., 2023). Guo et al. (2017) found that early neural networks tend to be overconfident and mitigated this by adjusting the temperature during the generation process. Subsequently, Desai and Durrett (2020) found that pre-trained Bert-style models have a relatively clear perception of their knowledge boundaries and Jiang et al. (2021) showed that the issue of overconfidence still persists in pre-trained language models. More recent studies have explored LLMs' perception of their knowledge boundaries. Kadavath et al. (2022) and Si et al. (2022) demonstrated LLMs can be reliable under approprite prompts. Kuhn et al. (2023) argued that the probability of generated tokens does not accurately reflect the probability of the generated answer and estimated LLMs' confidence in their answers based on the semantic consistency across multiple generations.

Verbalized Confidence. With the development of LLMs, some studies have shown that LLMs can express their confidence in answers in natural language (Lin et al., 2022; Yin et al., 2023; Tian et al., 2023; Xiong et al., 2023; Yang et al., 2023; Ni et al., 2024a). Lin et al. (2022) were the first to demonstrate that a model (GPT-3) can learn to express confidence about its answers using natural language. Recently, Yin et al. (2023) found that LLMs have difficulty in perceiving their knowledge boundaries and tend to be overconfident and Xiong et al. (2023) explored various methods of confidence extraction. To enhance LLMs' perception level, Tian et al. (2023) and Ni et al. (2024a) focused on prompting methods while Yang et al. (2023) proposed training methods.

Confidence Estimation via Internal States. LLMs' internal states have been found to contain information about their perception of the factuality of texts. (Azaria and Mitchell, 2023; Slobodkin et al., 2023; Chen et al., 2024) and (Wang et al., 2024) extended this approach to detect the factuality of model's self-generated content. This line of work utilized a shallow network (i.e., MLP) to

extract confidence from the hidden states of LLMs. Compared to prob-based methods, this tends to be more accurate because converting hidden states into token probabilities results in information loss. Additionally, compared to training LLMs to express better verbalized confidence, we just need to train a lightweight network, making it much more cost-efficient.

Current research on LLMs' perception of their knowledge boundaries through hidden states has two main limitations: 1) It lacks guidance on the timing for extracting these hidden states—whether before or after response generation. 2) It has not introduced methods for further enhancing this perception. Therefore, we try to tackle these two problems in this paper.

3 Preliminaries

In this section, we introduce the task formulation and the basic experimental setup.

Open-Domain QA. The goal of open-domain QA is for the model to provide a correct answer a to a given question q. LLMs store vast amounts of knowledge within their parameters and can directly provide answers without depending on any external resources. We only focus on instructing LLMs to output answers based on their internal knowledge and this can be described as follows:

$$a = f_{LLM}(q), \tag{1}$$

LLMs' Perception of Their Knowledge Boundaries. Recent works (Azaria and Mitchell, 2023; Chen et al., 2024; Wang et al., 2024; Slobodkin et al., 2023; Su et al., 2024) have demonstrated that the internal states of LLMs can be used to detect hallucinations in texts and contain information about LLMs' awareness of their knowledge boundaries.

The process of estimating the model's confidence based on its hidden states is as follows. For a given model M and the question q, it can gain the internal states $I_{M,q}$ and generate the response $a_{M,q}$ as:

$$I_{M,q}, a_{M,q} = f_M(q), \tag{2}$$

Then, we extract the confidence from the internal state $I_{M,q}$:

$$c_{M,q} = \mathcal{E}(I_{M,q}), \tag{3}$$

where $\mathcal{E}$ is the confidence extractor. $c_{M,q} = 1$ indicates the model is confident that it knows the correct answer while $c_{M,q} = 0$ means the verse vice. The confidence extractor $\mathcal{E}$ can be learned through a dataset $D_M^{train} = \{(I_{M,q_i}^{train}, c_{M,q_i}^{train})_{i=1}^n\}$ where n is the count of samples in this dataset. The ground-truth confidence c_{M,q_i}^{train} is set to 1 if the model can correctly answer the question q_i (i.e., $a_{M,q_i} = 1$); otherwise, it is set to 0.

Once $\mathcal{E}$ is learned, we can perform confidence estimation during inference.

Existing work often uses the hidden state of the last generated token to extract the model's confidence. In this paper, we extract representations both before and after answer generation to investigate whether the model can sense its knowledge boundaries prior to generating a response. The specific extraction method is detailed in section 4.1.

Dataset	Train	Dev	Test
NQ	10,000	6,489	3,610
HotpotQA	10,000	6,000	7,405
MMLU	7,021	3,510	3,511

Table 1: Count of samples for each dataset.

4 Confidence Estimator Training

In this section, we introduce how to train the confidence extractor, including the construction of unsupervised training data, the structure of the confidence extractor, and the specific training settings.

4.1 Unsupervised Training Data Generation

In Transformer-based models, the model performs next token prediction, where the generation of each token is based on the semantic vectors (i.e., internal states) of the preceding tokens in its sequence. For a give question q, suppose the input tokens are $\{q_1, q_2, \ldots, q_n\}$ and the tokens of answer a generated by a Transformer-based model are $\{a_1, a_2, \ldots, a_m\}$. The hidden states for generating each token of the answer are represented as:

$$I_a = \{I_{a_1}^l, I_{a_2}^l, \ldots, I_{a_m}^l\}_{l=1}^L, \tag{4}$$

where I_x^l represents the representation of tokens before x at layer l and L is the total number of layers in the model. Note that $I_{a_1}^l$ only contains the information about the question.

Layer Selection Previous work (Azaria and Mitchell, 2023) has found that the representations from the intermediate layers best capture the

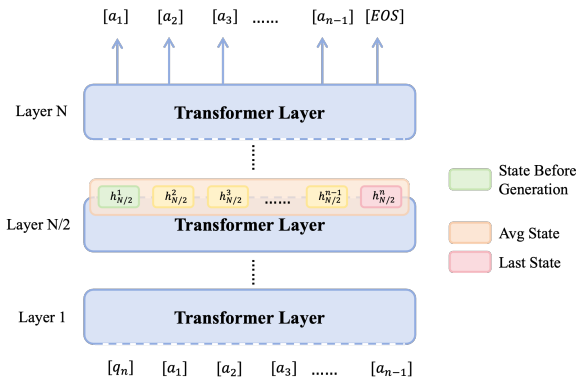


Figure 1: Hidden states extraction during generation.

model's awareness of factuality. Therefore, we extract representations from the intermediate layers (i.e., 16 for Llama2-Chat-7B) to construct the internal states I.

We aim to explore the timing for extracting hidden states. To investigate this, we construct I at two stages (See Figure 1) to examine the gap between the perception levels before and after generating a response:

Before Response Generation. We ask the model to generate answers for each question in the training set and extract the internal vector $I_{a_1}^{mid}$ which is used to generate the first answer token at the mid layer. For each question, we construct $\{I_{a_1}^{mid}, c\}$ as the training sample where $c = 1$ if the ground-truth answer in the response; otherwise, $c = 0$.

After Response Generation. Unlike pre-generation states, For the given question q, we construct the training sample $\{I, c\}$ in two ways:

1. **Last State**. We take the representation used to generate the last token of the answer $I_{a_m}^{mid}$ as I.

2. **Avg State**. We take the average of the the representations used to generate the answer $\frac{1}{m}\sum_{i=1}^{m} I_{a_i}^{mid}$ as I.

All the other settings are the same as those used in Pre-state before response generation.

4.2 Binary Confidence Estimator

Similar to previous works (Azaria and Mitchell, 2023; Chen et al., 2024; Su et al., 2024), we take a lightweight MLP (Multilayer Perceptron) network as the classifier. The classifier takes the internal states which are constructed as described in Section § 4.1 as input and outputs a binary confidence label indicating whether the model is confident to provide a correct answer. This process can be mathematically expressed as:

$$\hat{c} = \text{argmax}_{\theta}\left(\text{MLP}\left(\text{ReLU}(W \cdot I + b)\right)\right) \quad (5)$$

where $\hat{c}$ is the predicted confidence,θ is the softmax function, and I is the internal state vector with the shape of $1 \times dim_I$. W and b represent weights and the biases of MLP, respectively. We use a 4-layer MLP with the following number of hidden units in each layer: $(512 \rightarrow 64 \rightarrow 32 \rightarrow 2)$.

Training. We employ cross-entropy as the loss function, which is defined as follows:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} c_i \log(\hat{c}_i) \quad (6)$$

where c_i is the ground-truth confidence and N=2. We randomly initialize the model parameters and use the Adam optimizer with an initial learning rate of 5×10^{-5}. To enhance the reliability of the results, we train the model 30 epochs under three random seeds (0, 42, 100) and report the test results as the average performance across these three seeds. We randomly sample 1,000 correctly answered and 1,000 incorrectly answered examples from the training set to train this network.

Inference. During inference, we can determine whether a model is confident to provide a correct answer by using the required internal states as input to the MLP while the LLMs generate responses.

5 LLMs' Perception Before Response Generation

In this section, we evaluate the gap between LLMs' perception level before and after generating a response, as well as the impact of question difficulty, question format, and the amount of generated content on this level.

5.1 Experimental Setup

Datasets. We conduct experiments on three widely-used open-domain QA benchmark datasets, including Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (HQ) (Yang et al., 2018), and MMLU (Hendrycks et al., 2020). NQ and HotpotQA are two free-form QA datasets that primarily evaluate the model's factual knowledge, with varying levels of difficulty. NQ is constructed from Google Search queries, with annotated short and

Datasets	Metrics	Llama2-7B		Llama3-8B		Qwen2-7B		Llama2-13B	
		Vanilla	COT	Vanilla	COT	Vanilla	COT	Vanilla	COT
NQ	Acc	26.12	36.43	27.53	44.35	27.31	37.76	32.27	41.99
	Align-P	73.65	67.51	73.79	65.21	**72.69**	64.06	68.67	65.16
	Align-L	73.73	68.98	74.73	67.56	70.77	67.85	**72.15**	66.12
	Align-A	**74.82**	**70.12**	**75.35**	**67.66**	72.65	**69.78**	71.20	**67.03**
HotpotQA	Acc	19.93	29.55	21.63	36.79	24.96	33.34	23.69	33.10
	Align-P	79.69	74.36	78.58	74.61	**79.34**	**76.79**	75.91	73.55
	Align-L	79.16	**74.77**	**80.77**	74.59	75.13	75.20	**77.66**	**74.10**
	Align-A	**79.91**	72.71	80.61	**74.67**	77.77	75.33	76.32	72.57
MMLU	Acc	42.20	45.51	62.49	63.77	68.72	68.63	50.58	51.18
	Align-P	62.86	63.83	71.95	68.17	69.33	68.68	64.88	64.25
	Align-L	68.11	66.55	72.86	70.43	70.02	72.66	67.75	67.96
	Align-A	**68.71**	**67.95**	**73.98**	**71.61**	**72.57**	**72.74**	**69.18**	**69.30**

Table 2: The QA performance and LLMs' perception of their knowledge boundaries on NQ, HotpotQA, and MMLU datasets. Bold denotes the highest level for each model on each dataset. Align-P, Align-L, and Align-A represent the alignment scores for results when using the Pre-generation State, Last State, and Avg State, respectively.

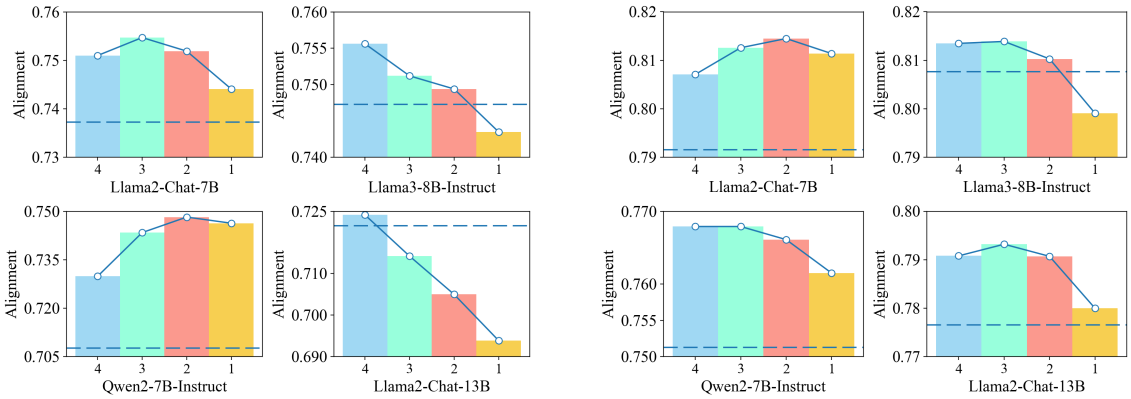
Datasets	Metrics	Vanilla					COT				
		Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓	Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓
NQ	Align-P	27.53	73.79	17.38	**8.02**	18.18	44.35	65.21	41.12	15.78	19.01
	Align-L	27.53	74.73	21.47	9.60	15.67	44.35	67.56	41.50	**14.79**	17.65
	Align-A	27.53	**75.35**	19.71	8.41	**16.23**	44.35	**67.66**	43.91	15.95	**16.39**
HQ	Align-P	21.63	78.58	26.91	13.35	8.08	36.79	74.61	31.29	**9.95**	15.44
	Align-L	21.63	**80.77**	24.88	11.24	7.99	36.79	74.59	35.18	11.90	13.51
	Align-A	21.63	80.61	24.55	**11.15**	8.24	36.79	**74.67**	37.72	13.13	**12.20**
MMLU	Align-P	62.49	71.95	67.83	16.36	**11.70**	63.77	68.17	80.24	23.87	7.97
	Align-L	62.49	72.86	63.39	**13.68**	13.46	63.77	70.43	75.33	**20.28**	9.29
	Align-A	62.49	**73.98**	64.95	13.90	12.12	63.77	71.61	76.97	20.51	**7.88**

Table 3: Detailed perception for Llama3-8B-Instruct. Align., Conf., Overcon., and Conserv. stands for Alignment, Confidence level, Overconfidence, and Conservativeness, respectively. Bold denotes the best scores on each dataset.

long answers. HotpotQA is a dataset consisting of question-answer pairs that require multi-hop reasoning. These pairs are collected via Amazon Mechanical Turk. MMLU is a multi-choice dataset containing questions from 57 different subjects, including STEM, humanities, and social sciences, with difficulty levels ranging from elementary to advanced. It is a benchmark designed to assess models' knowledge gained during pretraining. For the NQ dataset, we use its test set as our NQ-test, the validation set as NQ-dev, and sample 10000 examples from the training set as NQ-train. For HotpotQA, similar to previous work (Ni et al., 2024a), we use the validation set as the HQ-test. Additionally, we sample 10,000 examples from the training set as the HQ-train and 6,000 examples as HQ-dev. For MMLU, we randomly sample 50% of its test set as MMLU-train, and split the remaining portion equally into MMLU-dev and MMLU-test. Count of samples for each dataset can be found in Table 1.

Models. We conduct experiments on four representative open-source models, including Llama2-7B-Chat, Llama3-8B-Instruct, Qwen2-7B-Instruct, and Llama2-13B-Chat. We use half-precision for the 13B model. For all the models, we set the temperature to 1.0 and select the token with the highest probability at each position (i.e., greedy search). Unless otherwise specified, all the other parameters are set to their default values.

Metrics. Following previous research (Ni et al., 2024a), we use accuracy to evaluate the model's



(a) Alignment on NQ

(b) Alignment on HotpotQA

Figure 2: The performance of C^3 under different β when using Last State. We only adjust the original confidence when the model lacks confidence in the β reformatted questions. The horizontal axis number k represents $\beta = k$.

QA performance considering a response correct if it contains the ground-truth answer. For the model's perception level, we use alignment, overconfidence, and conservativeness as the evaluation metric. Alignment refers to the proportion of samples where the model's confidence matches its QA performance, serving as an indicator of the model's overall perception level. Overconfidence and conservativeness represent the proportions of samples where the model's confidence exceeds or falls below its actual capabilities, respectively, and are used to illustrate why the model's perception level is not perfect.

To study the impact of the information content in generated responses, we use two strategies to instruct the model to provide the answer: **Vanilla** where the model is simply asked to provide the correct answer and **COT** where the model is required to first output its reasoning process before giving the final answer (Wei et al., 2022).

We conduct experiments on free-form QA and multi-choice QA datasets to investigate the influence of task format and the comparison of performance on HQ and NQ datasets represents the effect of question difficulty. For the details of data construction and confidence estimation, you can find in section § 4.1.

5.2 Results and Analysis

The QA performance and alignment results for all the models can be found in Table 2, and the detailed perception results for Llama3-8B-Instruct can be seen in Table 3. Detailed results for the other models can be found in Appendix §A. We observe that: 1) **LLMs can perceive their knowledge boundaries before generating the response, and incorporating the representations of the generated answer further enhances the perception.** Across all the models and datasets, Align-P consistently achieves high perception level. Align-L and Align-A often show improvement compared to Align-P. This provides us a trade-off between judging whether the model can provide the correct answer and the computational cost. On one hand, if we determine that the model cannot provide the correct answer before it generates a response, we can stop the generation to save computational cost, especially in scenarios like CoT where responses tend to be long. On the other hand, making the judgment after generation improves the accuracy of the assessment.

2) **Including reasoning process in the output does not widen the gap between an LLM's perception level before and after generating a response, but it may reduce the model's overall perception level.** Experimental results show that COT does not significantly increase the gap between Align-P and Align-L (or Align-A) in any scenario. Additionally, in free-form QA tasks, COT often significantly improves QA performance but tends to reduce the model's perception level. From Table 3, it can be seen that COT leads to an increase in the model's confidence. However, this change in confidence does not align with the changes in QA performance, which may result increase the overconfidence or conservativeness, thereby harming the alignment. However, on MMLU, COT has no significant impact on QA performance (similar to

Models	Methods	NQ					HotpotQA				
		Acc	Align.↑	Conf.	Overconf.↓	Conserv.↓	Acc	Align.↑	Conf.	Overconf.↓	Conserv.↓
Llama2-7B	Vanilla	26.12	74.82	13.5	**6.28**	18.9	19.93	79.91	23.43	11.8	8.3
	RQ-2	21.91	76.17	17.73	9.82	14.01	17.58	80.31	18.51	10.31	9.38
	RQ-4	19.09	78.04	14.76	8.82	13.14	16.34	77.79	25.8	15.84	**6.38**
	RQ-6	15.46	79.64	20.45	12.68	7.68	13.8	**84.17**	12.04	**7.03**	8.8
	RQ-8	13.66	**83.2**	14.8	8.98	**7.83**	12.72	82.69	16.22	10.4	6.9
Llama3-8B	Vanilla	27.53	75.35	19.71	8.41	16.23	21.63	**80.61**	24.55	11.15	8.24
	RQ-2	26.2	75.9	26.15	12.02	12.08	19.81	80.43	19.93	**9.84**	9.72
	RQ-4	23.07	78.46	18.6	8.53	13.01	18.65	80.4	23.92	12.44	7.17
	RQ-6	22.22	**79.76**	17.8	**7.91**	12.33	17.8	80.5	23.75	12.73	6.78
	RQ-8	21.0	79.09	23.93	11.92	**8.98**	16.29	80.26	23.85	13.65	**6.09**
Qwen2-7B	Vanilla	27.31	72.57	26.76	13.4	13.95	24.96	77.79	27.18	12.23	10.0
	RQ-2	21.3	75.02	26.57	15.12	9.85	20.51	80.4	21.91	10.5	9.1
	RQ-4	20.08	**77.94**	22.4	**12.19**	9.87	16.85	80.65	20.42	11.46	7.89
	RQ-6	17.84	77.61	22.21	13.38	9.01	15.65	81.22	18.6	10.87	7.91
	RQ-8	18.12	77.42	22.77	13.62	**8.97**	14.75	**81.89**	17.34	**10.35**	**7.76**
Llama2-13B	Vanilla	32.27	71.2	29.16	12.84	15.96	23.69	76.32	27.27	13.63	10.05
	RQ-2	25.9	74.25	25.08	12.47	13.29	21.38	77.58	24.75	**12.9**	9.52
	RQ-4	22.74	76.21	21.52	11.28	12.5	18.83	78.52	24.33	13.49	7.99
	RQ-6	19.31	78.64	19.48	10.77	10.59	16.87	79.39	22.29	13.01	7.59
	RQ-8	16.95	**80.49**	18.05	**10.3**	**9.21**	14.79	**79.59**	24.29	14.96	**5.46**

Table 4: LLMs' QA performance and perception level on reformatted questions when using Avg State.

the previous findings (Sprague et al., 2024)), and its effect on perception level shows no clear pattern.

3) **LLMs exhibit better perception of their knowledge boundaries on more difficult questions.** The results from Table 2 and Table 3 show that, compared to NQ, the model's QA performance on HotpotQA is lower, yet its confidence remains at a comparable or even higher level. This reduces conservativeness and improves alignment. This is consistent with the previous findings in Ni et al. (2024b).

6 C^3: Consistency-based Confidence Calibration

Mitigating potential risks based on the model's judgment is crucial, especially in safety-critical domains like healthcare. When a model claims it can answer correctly but fails, it introduces risks. To address this, we propose C^3, a method focused on reducing overconfidence and further enhancing the model's awareness of its knowledge boundaries.

6.1 Methodology

LLMs should maintain consistent confidence in the knowledge they know. For an open-ended question, if the model initially believes it can answer correctly but then doubts its ability after the question's form changes, this indicates a lack of confidence and suggests that the previous judgment may be wrong (i.e., overconfident). Our proposed method is based on this assumption and consists of the following two steps:

Question Reformulation. For a given open-ended question, we first ask the model to generate k possible answers, and then we select α unique answers from them as options to construct the multi-choice question which we call this process as **RQ**-α (Question Reformulation-α). We set k to 10 and α to 2, 4, 6, and 8, respectively.

Confidence Calibration We aggregate the model's confidence across various multiple-choice questions to calibrate its confidence on the original question. The specific strategy is as follows: if the model believes it can answer an open-ended question correctly (i.e., $c_{oe} = 1$) but lacks confidence in β modified questions, we adjust c_{oe} to 0. β is a number between 0 and α. In this paper, we set $\beta = \alpha$ because this provides stable improvements.

Metrics. In addition to the existing metrics,we specifically focus on avoiding mistrust in wrong predictions, which refers to the extent to which potential risks are mitigated based on the model's confidence. This can be formulated as:

$$\text{Risk} = 1 - \frac{\sum_{i=1}^{n} \mathbb{1}(a_i = 0 \text{ and } c_i = 0)}{\sum_{i=1}^{n} \mathbb{1}(a_i = 0)}, \quad (7)$$

The rest of the experimental settings are the same as in Section §5.1.

Models	States	Methods	NQ				HotpotQA			
			Align.↑	Conf.	Risk↓	Overcon.↓	Align.↑	Conf.	Risk↓	Overcon.↓
Llama2-7B	Avg	Vanilla	74.82	13.50	8.5	6.28	79.91	23.43	14.74	11.80
		C^3	**75.55**	8.86	**4.86**	**3.59**	**80.70**	18.94	**11.44**	**9.16**
	Last	Vanilla	73.73	21.59	14.71	10.87	79.16	25.91	16.75	13.41
		C^3	**75.16**	14.23	**8.76**	**6.47**	**80.71**	21.56	**13.07**	**10.46**
Llama3-8B	Avg	Vanilla	75.35	19.71	11.6	8.41	80.61	24.55	14.24	11.15
		C^3	**75.69**	15.82	**8.69**	**6.30**	**81.27**	18.91	**10.21**	**8.00**
	Last	Vanilla	74.73	21.47	13.26	9.60	80.77	24.88	14.34	11.24
		C^3	**75.56**	15.37	**8.47**	**6.14**	**81.35**	18.89	**10.15**	**7.95**
Qwen2-7B	Avg	Vanilla	72.65	26.76	18.43	13.40	77.77	27.18	16.28	12.23
		C^3	**74.14**	20.03	**12.78**	**9.29**	**78.69**	21.80	**12.09**	**9.07**
	Last	Vanilla	70.77	29.41	21.54	15.66	75.13	29.95	19.88	14.93
		C^3	**72.99**	22.82	**15.49**	**11.26**	**76.79**	24.16	**14.94**	**11.21**
Llama2-13B	Avg	Vanilla	71.20	29.16	18.97	12.84	76.32	44.32	24.69	12.36
		C^3	**71.82**	17.76	**10.09**	**6.83**	**77.80**	22.42	**13.71**	**10.47**
	Last	Vanilla	72.15	26.92	16.61	11.25	77.66	25.10	15.55	11.88
		C^3	**72.41**	17.79	**9.68**	**6.56**	**79.08**	20.47	**11.6**	**8.85**

Table 5: The results of LLMs' perception level of their knowledge boundaries after calibration with C^3. Bold indicates the best scores for each state.

6.2 Results and Analysis

LLMs' QA performance and perception levels on reformatted questions can be seen in Table 4. The performance of C^3 can be seen in Table 5. We find that: 1) **LLMs can be misled by incorrect but related answers, leading to worse QA performance.** As the number of options increases, despite the higher likelihood of including the correct answer among them, LLMs' QA performance consistently decline. This indicates that the LLMs struggle to select the correct answer when faced with similar candidate answers. 2) **Question reformulation usually lead to alignment improvement.** The decline in LLMs' confidence on reformatted questions is often less than the decrease in QA performance, which reduces conservativeness and enhances alignment. More accurate assessments of these questions enable us to obtain reliable supplementary information. 3) C^3 **improves alignment without altering the generated answers, resulting in reduced risk in the response.** As shown in Table 5, our method consistently reduces LLMs' confidence in all the scenarios as expected, mitigating overconfidence, which leads to an increase in alignment and significantly improved the safety of the generated answers.

The impact of the threshold β in C^3. We demonstrate the impact of β on the calibration effect in C^3. The larger the β, the stricter the calibration rules, resulting in fewer calibrated samples. We conducted experiments based on the last states, and the results are shown in Figure 2. The experimental results show that when β is relatively large, it often yields good results. However, larger values are not always better, as an excessively large β may result in overly strict confidence modification criteria. The number of constructed questions and the modification standards require further research in the future.

7 Conclusion

In this paper, we first analyze LLMs' ability to assess the factuality of their answers based on internal representations of the question alone versus the entire question-answer sequence find that LLMs can predict the correctness of their responses before generation, offering a cost-efficient approach by avoiding inference, with this ability further refined after generation. However, in safety-critical areas, effectiveness is prioritized over efficiency. Therefore, we propose C^3 (Consistency-based Confidence Calibration) to further calibrate the model's perception level after response generation. Experimental results show that C^3 enhances the model's perception level without altering the output content and significantly mitigates the potential risks.

Limitations

First, we conduct research based on binary confidence to determine whether the model's output can be trusted. Further exploration is needed in the future for more fine-grained confidence. Second, we focus on the model's perception of its factual knowledge boundaries. The model's perception of its non-factual knowledge boundaries require further investigation. Third, due to resource limitations, we conduct our experiments only on 7B and 13B models. The effectiveness of our approach on larger models remains to be validated.

Ethics Statement

We approach ethics with great care. In this paper, all the datasets and models we use are open-source. Additionally, the methods we propose aim to enhance the reliability of LLMs' responses and do not encourage or induce the model to generate any harmful information.

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*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

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

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.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024a. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. *arXiv preprint arXiv:2402.11457*.

Shiyu Ni, Keping Bi, Lulu Yu, and Jiafeng Guo. 2024b. Are large language models more honest in their probabilistic or verbalized confidence? *arXiv preprint arXiv:2408.09773*.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un) answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448*.

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 preprint arXiv:2305.14975*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yanling Wang, Haoyang Li, Hao Zou, Jing Zhang, Xinlei He, Qi Li, and Ke Xu. 2024. Hidden question representations tell non-factuality within and across large language models. *arXiv preprint arXiv:2406.05328*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023. Alignment for honesty. *arXiv preprint arXiv:2312.07000*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

A Appendix

Datasets	Metrics	Vanilla					COT				
		Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓	Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓
NQ	Align-P	26.1	73.65	11.09	5.66	20.69	36.4	67.51	28.27	12.17	20.32
	Align-L	26.1	73.73	21.59	10.87	15.4	36.4	68.98	32.52	13.55	17.46
	Align-A	26.1	74.82	13.5	6.28	18.9	36.4	70.12	32.01	12.73	17.15
HQ	Align-P	19.9	79.69	16.47	8.42	11.88	29.55	74.36	27.46	11.78	13.86
	Align-L	19.9	79.16	25.91	13.41	7.43	29.55	74.77	28.74	12.21	13.02
	Align-A	19.9	79.91	23.43	11.8	8.3	29.55	72.71	34.83	16.29	11.0
MMLU	Align-P	42.2	62.86	33.7	14.06	23.08	45.51	63.83	37.21	14.26	21.91
	Align-L	42.2	68.11	31.4	10.28	21.61	45.51	66.55	38.36	13.47	19.98
	Align-A	42.2	68.71	32.13	10.35	20.94	45.51	67.95	34.04	10.61	21.44

Table 6: Detailed perception for Llama2-Chat-7B. Align., Conf., Overcon., and Conserv. stands for Alignment, Confidence level, Overconfidence, and Conservativeness, respectively. Bold denotes the best scores on each dataset.

Datasets	Metrics	Vanilla					COT				
		Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓	Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓
NQ	Align-P	27.31	72.69	0	0	27.31	37.76	64.06	29.3	13.74	22.2
	Align-L	27.31	70.77	29.41	15.66	13.56	37.76	67.85	34.27	14.33	17.82
	Align-A	27.31	72.65	26.76	13.4	13.95	37.76	69.78	37.67	15.07	15.15
HQ	Align-P	24.96	79.34	19.9	7.81	12.86	33.34	76.79	25.59	7.73	15.49
	Align-L	24.96	75.13	29.95	14.93	9.93	33.34	75.2	29.12	10.29	14.51
	Align-A	24.96	77.77	27.18	12.23	10.0	33.34	75.33	35.33	13.33	11.34
MMLU	Align-P	68.72	69.33	92.26	27.19	3.48	68.63	68.68	92.45	27.75	3.57
	Align-L	68.72	70.02	87.58	24.5	5.48	68.63	72.66	81.46	20.26	7.07
	Align-A	68.72	72.57	82.27	20.57	6.85	68.63	72.74	85.76	22.38	4.89

Table 7: Detailed perception for Qwen2-7B-Instruct. Align., Conf., Overcon., and Conserv. stands for Alignment, Confidence level, Overconfidence, and Conservativeness, respectively. Bold denotes the best scores on each dataset.

Datasets	Metrics	Vanilla					COT				
		Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓	Acc	Align.↑	Conf.	Overcon.↓	Conserv.↓
NQ	Align-P	32.27	68.67	18.91	8.98	22.35	41.99	65.16	24.72	8.78	26.06
	Align-L	32.27	72.15	26.92	11.25	16.6	41.99	66.12	35.47	13.67	20.2
	Align-A	32.27	71.2	29.16	12.84	15.96	41.99	67.03	35.6	13.29	19.69
HQ	Align-P	23.69	75.91	24.26	12.33	11.76	33.1	73.55	24.11	8.73	17.72
	Align-L	23.69	77.66	25.1	11.88	10.46	33.1	74.1	28.76	10.78	15.12
	Align-A	23.69	76.32	27.27	13.63	10.05	33.1	72.57	30.79	12.56	14.87
MMLU	Align-P	50.58	64.88	55.23	19.97	15.15	51.18	64.25	49.76	17.77	17.97
	Align-L	50.58	67.75	51.58	16.71	15.54	51.18	67.96	54.94	18.51	13.53
	Align-A	50.58	69.18	44.32	12.36	18.46	51.18	69.3	46.58	13.66	17.04

Table 8: Detailed perception for Llama2-Chat-13B. Align., Conf., Overcon., and Conserv. stands for Alignment, Confidence level, Overconfidence, and Conservativeness, respectively. Bold denotes the best scores on each dataset.