Teaching Large Language Models to Express Knowledge Boundary from Their Own Signals

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved great success, but their occasional content fabrication, or hallucination, limits their practical application. Hallucination arises because LLMs struggle to admit ignorance due to inadequate training on knowledge boundaries. We call it a limitation of LLMs that they can not accurately express their knowledge boundary, answering questions they know while admitting ignorance to questions they do not know. In this paper, we aim to teach LLMs to recognize and express their knowledge boundary, so they can reduce hallucinations caused by fabricating when they do not know. We propose COKE, which first probes LLMs' knowledge boundary via internal confidence given a set of questions, and then leverages the probing results to elicit the expression of the knowledge boundary. Extensive experiments show COKE helps LLMs express knowledge boundaries, answering known questions while declining unknown ones, significantly improving in-domain and out-of-domain performance.

1 Introduction

Large language models (LLMs) have emerged as an increasingly pivotal cornerstone for the development of artificial general intelligence. They exhibit powerful intellectual capabilities and vast storage of knowledge (Brown et al., 2020; Ouyang et al., 2022; Achiam et al., 2023), which enables them to generate valuable content. Recent research demonstrates that LLMs excel in passing various professional examinations requiring expert knowledge in domains like medical (Jin et al., 2021) and legal (Cui et al., 2023). Nevertheless, human users are hardly willing to seek professional suggestions from LLMs, due greatly to **hallucinations** in LLMs. Hallucinations in LLMs refer to the phenomenon that existing LLMs frequently generate untruthful information (Zhang et al., 2023b; Ji et al., 2023), which greatly undermines people's trust and acceptance of LLM-generated content.

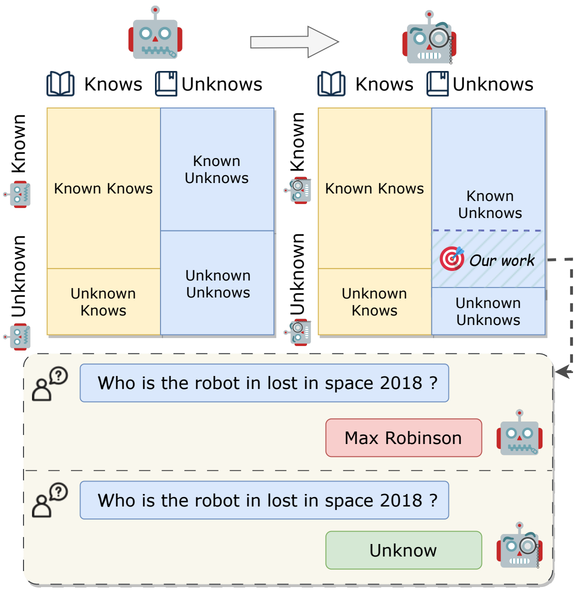

Figure 1: The evolution of the Known-Unknown Quadrant. The yellow portion represents the model's parametric knowledge. Our method increases the "Known Unknows", helping the model recognize and articulate its knowledge limitations.

An important cause of hallucinations is the model's insufficiency in knowledge boundary expression, which originates from the learning paradigm of LLMs. Pre-training and instruction fine-tuning serve as the two indispensable learning stages for current LLMs. The learning mechanism of these stages is to encourage LLMs to generate the provided text, which also makes LLMs prone to fabricating content when LLMs do not possess relevant knowledge (joh, 2023; Gekhman et al., 2024). Hence, LLMs are hardly instructed to express their ignorance, which is a lack of accurate knowledge boundary expression. Given a specific LLM and a question set, the corresponding question-answer pairs can be categorized based on two factors: (1) whether the model has corresponding parametric knowledge (knows v.s. unknows), and (2) whether the model is aware of the first factor (known v.s. unknown), as is depicted in Figure 1. Hallucinations frequently occur in the "Unknown Unknows" scenarios, where the model is unaware that it should explain its ignorance like humans, instead of struggling to give a hallucinated response.

Fine-tuning models to express knowledge boundaries faces two significant challenges. The first challenge is how to efficiently obtain data that reflects the internal knowledge of a specific model. Even if evaluation questions are easy to construct, obtaining expert-level answers in certain fields is costly. Additionally, since the model might produce correct answers in different forms from the reference answers, evaluating their correctness is also challenging (Kadavath et al., 2022; Zou et al., 2023). The second challenge is enabling the model to express its knowledge boundary robustly (Ren et al., 2023). We expect consistent knowledge boundary expression across prompts and generalization across domains.

To address the above two challenges, we propose COKE, an **Co**nfidence-derived **K**nowledge boundary **E**xpression method which teaches LLMs to express knowledge boundaries and decline unanswerable questions, leveraging their internal signals. Our method consists of two stages: a probing stage and a training stage. In the probing stage, we use the model's internal signals reflecting confidence to distinguish between answerable and unanswerable questions, avoiding reliance on external annotations. This allows for easy collection of large data and avoids conflicts between the model's internal knowledge and annotations. In the training stage, we construct prompts for each question using three representative types: prior awareness, direct awareness, and posterior awareness. Then, we apply regularization by incorporating the squared differences in confidence across different prompts for the same question into the loss function to enhance consistency. This training setup helps the model semantically learn to express knowledge boundary better, thereby enhancing its generalization ability.

To evaluate the model's knowledge boundary expression capability, we design an evaluation framework that comprehensively assesses the model's performance in both "knows" and "unknows" scenarios. We conduct extensive experiments on both in-domain and out-of-domain datasets. Results show that the model learns to use internal signals to help express knowledge boundary. Compared to directly using model signals for determination, the models trained with our method demonstrate better performance and generalization.

In summary, our contributions are:

- We explore the effectiveness of internal model signals in indicating confidence and demonstrate the model can learn to use its signals to express its knowledge boundaries after training.
- We propose a novel unsupervised method that leverages internal model signals and multi-prompt consistency regularization to enable the model to express its knowledge boundary clearly.
- We develop a framework for evaluating a model's ability to express its knowledge boundary, and experimental results demonstrate that the model can learn signals about the confidence of its knowledge and articulate its knowledge boundary.

2 Related Work

2.1 Knowledge Boundary Perception

While models are equipped with extensive parametric knowledge, some studies indicate their inability to discern the knowledge they possess from what they lack, thus failing to articulate their knowledge boundary (Yin et al., 2023; Ren et al., 2023). In terms of enhancing a model's awareness of its knowledge boundary, efforts can be categorized into two parts: one focuses on enabling the model to fully utilize its inherent knowledge, thereby shrinking the ratio of the model's "Unknown Knows" (Wei et al., 2022; Li et al., 2023; Tian et al., 2024). The other part focuses on enabling the model to acknowledge the knowledge it lacks, thereby reducing the ratio of the model's "Unknown Unknows". R-tuning (Zhang et al., 2023a) uses labeled data to judge the correctness of model responses and trains the model using the SFT method. Yang et al. (2023) and Kang et al. (2024) explore training methods based on RL. Focused on this aspect, our work investigates how to enable models to express knowledge boundaries without annotated data, while also considering consistent knowledge boundary expression across prompts and generalization across domains.

2.2 Uncertainty-based Hallucination Detection

Some work on hallucination detection focuses on obtaining calibrated confidence from LLMs. One segment of work involves utilizing the information from these models to compute a score that signifies

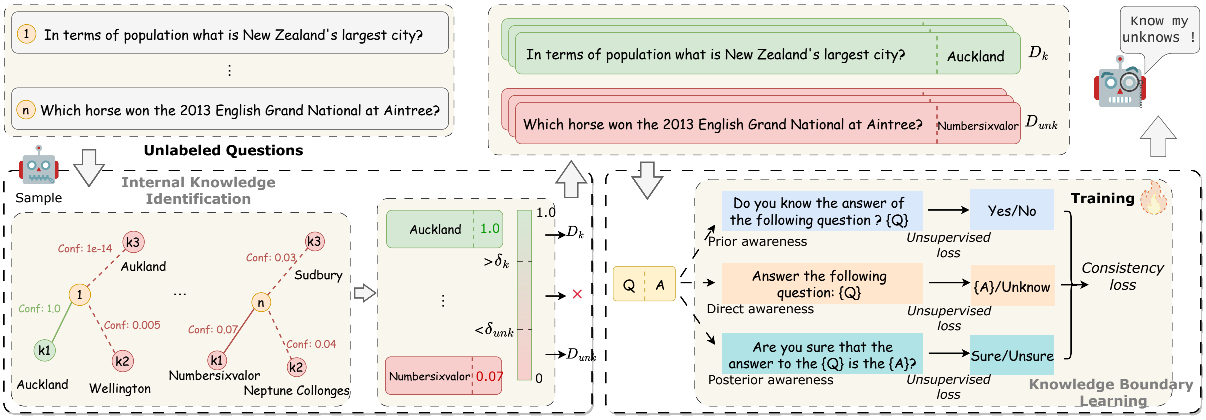

Figure 2: The procedure of CoKE, which consists of two stages. In the first stage, the model makes predictions for unlabeled questions. We obtain two parts, D_k and D_{unk}, based on the model confidence. In the second stage, we train with different prompts for the same question and use unsupervised loss and consistency loss to teach the model to express the knowledge boundary.

the model's uncertainty about knowledge (Manakul et al., 2023; Kuhn et al., 2023; Varshney et al., 2023; Duan et al., 2024). Another segment of work seeks to enable the model to express verbalized uncertainty (Lin et al., 2022; Xiong et al., 2023; Tian et al., 2023). Our work concentrates on enabling the model to explicitly express whether it is capable of answering, rather than generating a probability score. By allowing the model to express its knowledge boundary autonomously, users no longer need to concern themselves with detecting hallucinations, such as by setting uncertainty thresholds.

3 Knowledge Boundary Expression

3.1 Problem Formulation

We focus on exploring LLMs' capacity to perceive their internal knowledge. For a series of questions $Q = \{q_1, q_2, \ldots, q_n\}$, we categorize the questions based on whether the model has the knowledge required to answer them into two parts: questions that can be answered Q_k and questions that cannot be answered Q_{unk}. To minimize the interference from the model's reasoning ability, the questions used for testing the model are all single-hop questions that inquire about factual knowledge. For a given question q, the model M generates a prediction based on its parameter knowledge K_θ, represented as $y = M(K_\theta, q)$. We measure the model's awareness of its knowledge from two aspects: the awareness of the knowledge it possesses and the knowledge it does not possess. The former is represented as the ratio of the model's "Know Knows" to "Knows", denoted as R_k, while the latter is represented as the ratio of the model's "Know Unknows" to "Unknows", denoted as R_{unk}. Given a question $q \in Q_k$, R_K is set to 1 if the model's response y aligns with the knowledge k, and to 0 if the model either expresses uncertainty or provides an incorrect answer. For a question where $q \in Q_{unk}$, R_{unk} is assigned 1 if the model expresses uncertainty, and 0 if it fabricates an incorrect answer. We evaluate the model's awareness of its knowledge by testing on two types of q and calculating $S_{aware} = \frac{1}{2}(R_k + R_{unk})$. The model's awareness of its knowledge is more accurate as S_{aware} approaches 1, and less accurate as it approaches 0.

3.2 Method

Our insight is that the learning mechanism of LLM enables the model to search for the nearest knowledge k in its parameters as the answer to the query q. Although training allows the model to measure distances accurately, it does not teach it to refuse to answer based on the distance. Therefore, we hope the model can learn to use its signals to recognize when a large distance indicates a lack of knowledge to answer q. Our method involves two steps as shown in Figure 2: First, we use the model's own signals to detect knows and unknows; Second, we guide the model to learn these signals through instruction tuning, enabling it to express its knowledge boundary clearly.

3.2.1 Internal Knowledge Identification

To identify whether the model possesses the knowledge required to answer question q, we calculate the model's confidence about its prediction. The confidence of the model's prediction serves as a measure of the distance between query q and knowledge k. On the unlabeled question set Q, we let model M generate phrase-form predictions for each question. We only consider the distance between query q and the closest prediction; therefore, we use greedy decoding to obtain the prediction.

We use three model signals to represent the model's confidence: Min-Prob, Fst-Prob, and Prod-Prob. Min-Prob denotes the minimum probability among the m tokens that make up the model's prediction, $c = min(p_1, p_2, ..., p_m)$. Fst-Prob and Prod-Prob respectively represent the probability of the first token in the prediction and the product of all probabilities. Two conservative thresholds, δ_k and δ_{unk}, are established to decide whether the model has enough knowledge to answer a question. For questions with c below the threshold δ_{unk}, indicating the model is fabricating an answer due to insufficient knowledge, we define this subset as $D_{unk} = \{(q_i, y_i, c_i) \mid c_i < \delta_{unk}\}$ and use it to train the model to express its lack of knowledge. For questions with c above the threshold δ_k, indicating the model possesses the necessary knowledge, we define this subset as $D_k = \{(q_i, y_i, c_i) \mid c_i > \delta_k\}$ and use it to train the model to express that it knows the answer with increased confidence.

3.2.2 Knowledge Boundary Expression Learning

We guide the model in learning to express its knowledge boundaries clearly based on its own signals through instruction tuning. We believe that the model's expression of knowledge boundary awareness should possess two properties: honesty and consistency. Honesty requires the model to express whether it knows the answer to a question based on its certainty about the knowledge. For instance, it should not answer "I don't know" to questions it is certain about. For honesty, we fine-tune the model on the dataset obtained in the first step, enabling the model to admit its ignorance on D_{unk} and maintain its answers on D_k. Consistency requires the model to have the same semantic expression about whether it knows the same knowledge under different prompt formulations.

For consistency, we consider three different prompts for knowledge boundary awareness inquiries, which we refer to as prior awareness, direct awareness, and posterior awareness (Ren et al., 2023). **Prior awareness** involves the model assessing its ability to answer a question before actually providing an answer, with prompts like "`Do you know the answer to the question 'panda is a national animal of which country' honestly?`". **Direct awareness** involves the model responding directly to a query, supplying the answer if it possesses the knowledge, and admitting ignorance if it doesn't, with prompts like "`Answer the question 'panda is a national animal of which country'`". **Posterior awareness** involves the model's capacity to evaluate the certainty of its answers, with prompts like "`Are you sure that the answer to the 'panda is a national animal of which country' is 'China'`".

We hope that the model can express the same knowledge boundary under different prompts for the same question. It means that if the model determines that it possesses the knowledge under the prompt of prior awareness, it should be able to provide the answer when queried, and express confidence in its response when reflecting upon its answer. We teach the model to recognize its knowledge boundary by constructing three types of prompts for the same question. We incorporate the difference in probabilities of identical semantic responses under various prompts into the loss function, thereby ensuring the model's consistency across different prompts. Specifically, the loss function is defined as a combination of two components: L_{unsup}, which captures the discrepancy between the model's expression and the labels generated by its internal signals, and L_{con}, which ensures consistency of identical responses under different prompts:

$$L_{unsup} = -\sum_{1 \leq i \leq 3} \log P(y_i|x_i) \quad (1)$$

$$L_{con} = \sum_{1 \leq i,j \leq 3} \|P(y_i|x_i) - P(y_j|x_j)\|^2 \quad (2)$$

$$L = L_{unsup} + L_{con} \quad (3)$$

Previous research emphasizes that the MLP layer is a key component for storing knowledge in the transformer architecture LLM (Geva et al., 2021; Meng et al., 2022; Dai et al., 2022). Guided by these insights, we only fine-tune the weight matrix of the attention layer using LoRA (Hu et al., 2022). This strategy allows us not to change the internal knowledge of the model, but just let the model learn to express the of knowledge boundary based on the

Model	Method	TriviaQA			NQ			PopQA		
		K_{aware}	U_{aware}	$\text{S}_{\textbf{aware}}$	K_{aware}	U_{aware}	$\text{S}_{\textbf{aware}}$	K_{aware}	U_{aware}	$\text{S}_{\textbf{aware}}$
Llama2-Chat-7B	Orig.	100	0	50.0	100	0	50.0	100	0	50.0
	Fine-tune	93.9	6.2	50.1	88.6	3.1	45.8	93.5	1.9	47.7
	IDK-FT	80.8	78.0	79.4	45.5	87.6	66.6	62.8	83.6	73.2
	Uncertainty-Based									
	Min-Prob	61.8	86.2	74.0	33.4	91.4	62.4	57.7	89.3	73.5
	Fst-Prob	74.6	69.8	72.2	51.5	79.1	65.3	65.1	82.6	73.9
	Prod-Prob	68.3	81.2	<u>74.8</u>	45.8	87.0	<u>66.4</u>	63.7	86.4	<u>75.1</u>
	Prompt-Based									
	Prior	96.3	7.5	51.9	97.0	10.3	53.6	65.4	31.8	48.6
	Posterior	70.5	57.9	64.2	62.7	55.6	59.1	31.6	82.8	57.2
	IC-IDK	86.4	25.8	56.1	53.6	65.1	59.3	42.3	85.3	63.8
	Verb	14.3	95.8	55.1	17.5	95.0	56.3	17.6	97.3	57.4
	CoKE	76.1	74.0	**75.0**	56.0	84.2	**70.1**	71.1	83.0	**77.0**
Llama2-Chat-13B	Orig.	100	0	50.0	100	0	50.0	100	0	50.0
	Fine-tune	96.7	7.1	51.9	95.0	2.8	48.9	95.7	2.9	49.1
	IDK-FT	82.5	81.6	82.0	53.9	84.6	69.3	65.4	82.0	73.6
	Uncertainty-Based									
	Min-Prob	91.6	44.5	<u>68.1</u>	88.1	43.4	65.8	84.6	57.2	<u>70.9</u>
	Fst-Prob	92.9	34.1	63.5	90.6	30.7	60.7	87.4	51.0	69.2
	Prod-Prob	65.8	80.9	**73.3**	59.1	75.5	<u>67.3</u>	57.6	81.7	69.6
	Prompt-Based									
	Prior	88.6	14.2	51.4	81.3	26.5	53.9	38.2	81.8	60.0
	Posterior	100	0.30	50.0	100	0.0	50.0	100	0.10	50.0
	IC-IDK	99.7	1.5	50.6	96.8	6.7	51.7	90.8	25.1	58.0
	Verb	60.0	68.9	64.4	44.7	89.8	67.3	50.8	81.8	66.3
	CoKE	71.6	74.9	**73.3**	68.3	70.2	**69.2**	70.1	82.6	**76.4**

Table 1: Comparison of the performance of our method and the baseline method across an in-domain dataset (TriviaQA) and out-of-domain datasets (NQ and PopQA). We present results on two model scales: Llama2-Chat-7B and Llama2-Chat-13B.

Metric	Definition
K_{aware}	Proportion of *correct answers* on T_k
U_{aware}	Proportion of *expressions of unknown* or *correct answers* on T_{unk}
S_{aware}	$\frac{1}{2}(K_{aware} + U_{aware})$

Table 2: Knowledge awareness metrics.

confidence of the knowledge.

4 Experimental Setup

Datasets We consider three open-domain QA datasets: TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and PopQA (Mallen et al., 2023). These datasets are broad-coverage, knowledge-intensive QA datasets, making them well-suited for evaluating LLMs' capacity to perceive their internal knowledge. We utilize the train set of TriviaQA as our training data, treating it as unsupervised data by not using the labels. Natural Questions and PopQA serve as the out-of-domain test sets since they were not involved during the training process.

Metrics As mentioned in the Section 3.1, we evaluate the model's awareness of its knowledge from two aspects: the awareness of the knowledge it possesses and the awareness of the knowledge it does not possess. Since we cannot directly access the model's internal knowledge K_θ, we divide the test sets into two parts based on whether the model's predictions match the groundtruth: T_k represents the "Known Knows" of the model; T_{unk} contains both the "Unknown Unknows" and "Unknown Knows" cases. We expect the model to maintain correct answers on T_k, representing the retention of the "Known Knows" area of the model. At the same time, we expect the model to either express unknown on T_{unk}, signifying a reduction in the "Unknown Unknows" area, or provide correct answers, representing a decrease in the "Unknown Knows" area. We define the evaluation metrics as

Method	TriviaQA				NQ				PopQA			
	Brier↓	ECE↓	smECE↓	AUROC↑	Brier↓	ECE↓	smECE↓	AUROC↑	Brier↓	ECE↓	smECE↓	AUROC↑
Fst-Prob	0.29	0.31	0.20	0.79	0.36	0.45	0.25	0.73	0.29	0.38	0.22	0.83
Prob-Prob	0.38	0.42	0.23	**0.83**	0.55	0.65	0.31	0.73	0.46	0.57	0.28	**0.85**
Min-Prob	**0.24**	**0.26**	**0.19**	**0.83**	**0.29**	**0.39**	**0.23**	**0.77**	**0.25**	**0.34**	**0.20**	**0.85**

Table 3: Calibration results for different internal signals in Llama2-Chat-7B on TriviaQA, NQ, and PopQA.

shown in Table 2.

Baselines We consider two different types of baselines: uncertainty-based methods (white-box) and prompt-based methods (black-box). We also compared the original model (Orig.), the model fine-tuned with questions and their label (Fine-tune), and the model fine-tuned with question-label pairs, where responses to unknown questions are replaced by "Unknow" (IDK-FT). See Appendix A for more details.

Uncertainty-based methods **directly** use the model's internal signals to determine its self-awareness. The model's response consists of multiple tokens, and we experimented with three types of methods to calculate the final confidence score from the probabilities of these tokens:

- **Min token probability (Min-Prob)**: Use the smallest token probability in the model's prediction as the confidence score.
- **Product token probability (Prod-Prob)**: Use the product of the probabilities of all tokens in the model's prediction as the confidence score.
- **First token probability (Fst-Prob)**: Use the probability of the first token in the model's prediction as the confidence score.

Prompt-based methods use prompts to let models express their own knowledge boundary in natural language.

- **Prior prompt**: Similar to Ren et al. (2023) evaluating whether the model gives up on answering, we use the prompt to directly ask the model if it knows the answer to the question.
- **Posterior prompt**: Kadavath et al. (2022) shows the model can evaluate the certainty of its answers. We use the prompt to ask the model about the certainty of its answers.
- **In-context IDK (IC-IDK)**: Following Cohen et al. (2023), by integrating demonstrations into the prompt, we enable the model to express its knowledge boundary through in-context learning.
- **Verbalize uncertainty (Verb)**: Resent work (Tian et al., 2023) suggests that LLMs' verbalized uncertainty exhibits a degree of calibration. We let the model output verbalized uncertainty, and search for the optimal threshold in the training set.

5 Results and Analysis

5.1 Overall Performance

We present our main results on the in-domain and out-of-domain datasets in Table 1. Generally, we have the following findings:

Across all settings, we outperform prompt-based methods by a large gap. On Llama2-Chat-7B, CoKE obtains an S_{aware} of 75.0 compared to ≤ 64.2 by prompt-based methods on TriviaQA, and obtains an S_{aware} of 77.0 compared to ≤ 63.8 by prompt-based methods on PopQA. Models struggle to accurately express knowledge boundaries when it comes to the prior prompt, in-context learning, and posterior prompts. Meanwhile, models can express verbalized uncertainty through prompts, and their accuracy improves with larger models, but remains limited for models with fewer than 13 billion parameters. Interestingly, while accuracy improves with larger model sizes, self-awareness does not show significant gains in most cases. We believe that this capability may require even larger models to become evident.

Compared to uncertainty-based methods, CoKE can outperform in most settings. This demonstrates that CoKE enables the model to effectively learn its confidence signals and generalize beyond the training signals. On out-of-domain datasets, CoKE significantly outperforms uncertainty-based methods, indicating that thresholds derived from a dataset have poor transferability, while CoKE exhibits better generalization.

Compared to methods requiring labeled data for fine-tuning, CoKE demonstrates better generalization. Although CoKE performs worse than IDK-FT on in-domain test sets, it significantly out-

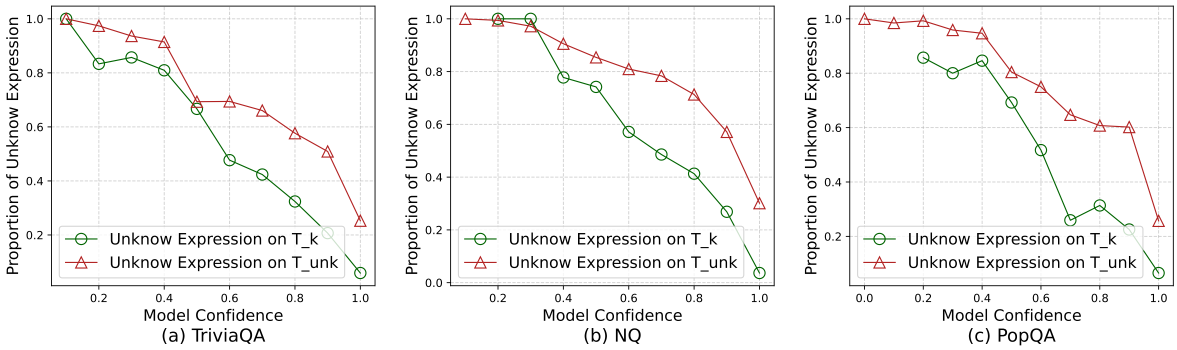

Figure 3: Model's "Unknow" expression ratio in question groups under different confidence scores (using minimum token probability). As the model's confidence score decreases, the ratio of "Unknow" expressions increases. The model exhibits a higher "Unknow" expression ratio on T_{unk} compared to T_k.

Training Signal	TriviaQA	NQ	PopQA
Fst-Prob	74.9	69.3	76.2
Prod-Prob	73.9	69.8	76.3
Min-Prob	**75.0**	**70.1**	**77.0**

Table 4: Different signals serve as the model's confidence score in training the expression of knowledge boundary. The metric is represented by the S_{aware}.

performs this supervised fine-tuning approach on out-of-domain datasets. This indicates that by leveraging the model's internal signals to teach LLMs to express knowledge boundaries, CoKE not only avoids reliance on labeled data but also achieves better generalization.

5.2 Effectiveness of Model Signals

We demonstrate the effectiveness of model internal signals in reflecting the model's knowledge boundaries through an evaluation of these signals. We used the same metrics as (Ulmer et al., 2024), including Brier score (BRIER, 1950), expected calibration error (ECE; Pakdaman Naeini et al., 2015), and smooth ECE (smECE; Blasiok and Nakkiran, 2024) to evaluate the model signals' calibration ability, and used AUROC to measure the model's ability to identify questions it doesn't know. As shown in Table 3, model internal signals perform poorly in terms of calibration, with high Brier and ECE scores. However, model internal signals perform well in determining whether the model is ignorant, with high AUROC scores, which is also reflected in the uncertainty-based methods in Table 1. By employing strict thresholds, our method mitigates signal noise while leveraging the signals' ability to discriminate between knowledge and ignorance.

We also analyze the effectiveness of different internal signals as training signals. As a training signal, the use of the minimum probability of multi-token outperforms other signals on both in-domain and out-of-domain datasets, as illustrated in Table 4. We consider that the minimum probability of multi-token is more easily mastered by the model. We leave the discovery of better signals reflecting the model's knowledge boundary and the utilization of multi-signal training for future work.

5.3 Leverage Internal Signals for Knowledge Boundary Expression

We investigated how our model utilizes confidence scores to express its knowledge boundary. Figure 3 illustrates the relationship between confidence scores and the model's tendency to respond with "Unknow". The results show a clear pattern: the model rarely answers "Unknow" at high confidence levels, while frequently doing so at low confidence levels. For example, with confidence scores below 0.4, the model almost always responds "Unknow", whereas it confidently provides answers when scores approach 1.0. This demonstrates that **the model effectively uses confidence scores to delineate its knowledge boundaries and generalizes well to out-of-domain data.**

Interestingly, we observed that for the same confidence level, the model responds "Unknow" more frequently to questions in T_{unk} compared to T_k. This suggests that **the model has learned to utilize additional implicit information beyond just the confidence score, which helps mitigate the problem of overconfidence in incorrect answers.** By incorporating the model's confidence as a supervisory signal during training, we reduce the noise associated with using minimum token probabil-

Method	$\mathbf{T_k}$				$\mathbf{T_{unk}}$			
	Correct (↑)	IDK (↓)	Wrong (↓)	Probs	Correct (↑)	IDK (↑)	Wrong (↓)	Probs
Orig.	100	0	0	0.86/ - / -	0	0	100	- / - /0.58
Min-Prob	61.8	38.2	0	0.98/0.68/ -	0	86.2	13.8	- /0.53/0.96
Posterior	70.5	29.5	0	0.86/0.85/ -	0	57.9	42.1	- /0.55/0.63
CoKE	76.1	22.3	1.6	0.92/0.68/0.60	3.7	70.3	26.0	0.64/0.52/0.75

Table 5: Percentage distribution of Llama-Chat-7B outputs on TriviaQA across three categories: correct answers, expressions of unknowns, and wrong answers. "Prob" represents the average min-probability for each category.

ity alone, resulting in improved performance compared to methods based solely on uncertainty.

5.4 Consistency of Knowledge Boundary Expression

We investigate the benefits of teaching a model to express knowledge boundary by using the strategy of constructing different prompts for the same question and applying a consistency regularization loss function. By adopting this strategy, we discover that it not only improves the model's ability to generalize, but also ensures a consistent expression of knowledge boundary under different prompts. Results from Table 6 indicate that the application of consistency loss, despite causing a slight decrease in S_{aware} on the in-domain dataset, leads to substantial improvements on the out-of-domain dataset, thereby demonstrating enhanced generalization. We also reported the consistency of the model's expression of knowledge boundary under different prompts, as shown in Table 6. We evaluate the model's consistency by randomly sampling two different types of prompt templates from prompt pools (see Appendix B.2). We notice that the model adopted with consistency loss is capable of expressing consistent knowledge boundaries for most questions under different prompts.

Method	TriviaQA		NQ		PopQA	
	S_{aware}	Con.	S_{aware}	Con.	S_{aware}	Con.
orig.	50.0	35.2	50.0	22.2	50.0	39.3
CoKE	75.0	**92.1**	**70.1**	**90.9**	**77.0**	**89.6**
w/o *Con-loss*	**75.6**	46.3	69.2	36.7	74.8	43.6

Table 6: The consistency of knowledge boundary expressions under different prompts. "Con." refers to the percentage of consistent responses when the model is presented with the same question using different prompt templates.

5.5 Error Analysis

Enhancing a model's self-awareness capability involves a tradeoff between maintaining performance on known knowledge (K_{aware}) and refusing to answer on unknown knowledge (U_{aware}). We analyze the outputs of CoKE and other methods, examining the types and proportions of different outputs within T_k and T_{unk}. As shown in Table 3, for the T_k portion, CoKE is able to maintain correct expressions for most questions, and the performance drop is due to the model becoming more conservative, refusing to answer some low-confidence questions. In the T_{unk} portion, the model correctly refuses to answer most questions it doesn't know, but issues of overconfidence still exist. Additionally, some originally correct answers become incorrect, and some originally incorrect answers become correct, which might result from the model changing its responses to questions with low confidence. Observing the average probabilities across different output types, Posterior methods show nearly identical probabilities for different outputs, while CoKE demonstrates a clearer alignment between its expression and answer confidence.

6 Conclusion

In this paper, we target the knowledge boundary expression problem and propose CoKE, a novel unsupervised approach for this task. Our approach is built on detecting signals of the model indicating confidence, and teaching the model to use its signals to express knowledge boundary. Through comprehensive experiments on in-domain and out-of-domain datasets, we show that our method can teach the model to use its signals, significantly enhancing the model's ability to accurately express knowledge boundary. Our work can be extended by seeking more internal signals that better reflect the model's confidence and exploring how to combine these signals to train the model, inspiring further research into models autonomously improving their ability to express knowledge boundaries without human annotations.

Limitations

We note three limitations of our current work. First is the accuracy of the evaluation methods. Because of the lack of a method to discover the internal knowledge of the model, we divided T_k and T_{unk} based on whether the model's answer matches the groundtruth, ignoring the impact of the model's erroneous beliefs. Another limitation is that to prevent exposure bias and the influence of multiple pieces of knowledge, we focused on the expression of knowledge boundary under short-form answers, without investigating the issue of long-form generation. Last, we focused on the model's ability to express the boundary of its internal knowledge, not extending to scenarios like self-awareness with external knowledge (e.g., RAG scenarios) or reasoning abilities (e.g., mathematics or logical reasoning).

Ethical Statement

We hereby acknowledge that all authors of this work are aware of the provided ACL Code of Ethics and honor the code of conduct.

Risks We propose COKE, which teaches models to express their knowledge boundaries using internal signals, thereby reducing hallucinations caused by fabricating answers when they do not know. Our experiments demonstrate that our method significantly reduces the instances of models fabricating answers to unknown questions. However, models may still occasionally produce fabricated answers in certain scenarios. Therefore, in practical applications, it is important to note that our method does not completely eliminate hallucinations, and there remains a risk of models generating fabricated content. Caution is advised in fields with stringent requirements.

References

2023. John schulman - reinforcement learning from human feedback: Progress and challenges.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jaroslaw Blasiok and Preetum Nakkiran. 2024. Smooth ECE: Principled reliability diagrams via kernel smoothing. In *The Twelfth International Conference on Learning Representations*.

GLENN W. BRIER. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640, Singapore. Association for Computational Linguistics.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. Association for Computational Linguistics.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *arXiv preprint arXiv:2405.05904*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea

Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *arXiv preprint arXiv:2403.05612*.

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*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, pages 41451–41530. Curran Associates, Inc.

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

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. 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.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372. Curran Associates, Inc.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Dennis Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Oh. 2024. Calibrating large language models using their generations only. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15440–15459, Bangkok, Thailand. Association for Computational Linguistics.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

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*.

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

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023a. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, and 1 others. 2023b. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zeyuan Allen Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Methodology

In this section, we elaborate on the rationale for selecting the baseline methods in our work, as well as the implementation details.

A.1 Uncertainty-based Methods

Inspired by works on uncertainty estimation for LLMs, we believe that confidence calculated through the model's internal signals can effectively reflect the model's self-awareness. Since we control the model to output only answer phrases instead of full sentences through prompting, we do not need to perform additional extraction on the generated content (Varshney et al., 2023; Duan et al., 2024), but instead directly compute using the logits of the tokens in the generated answer phrase.

In this work, we consider three methods for calculating the model's confidence using its internal signals:

- **Min token probability & Product token probability:** Varshney et al. (2023) found that the minimum and product of the probabilities of tokens that form important concepts in a model-generated sentence can effectively reflect the model's uncertainty. For Min token probability, we directly take the smallest probability among the tokens that compose the model-generated phrase as the model's confidence. For Product token probability, we calculate the product of the probabilities of each token, and then normalize it by the length to obtain the final confidence score.
- **First token probability:** Considering that the model may store the entire concept's information in the hidden state of the token at the beginning of the concept phrase (Zhu and Li, 2023), we use the probability of the first token to represent the confidence of the entire response.

To directly use the confidence score to predict the model's knowledge boundary, we determine whether the model expresses uncertainty based on whether the score exceeds a threshold. We determine the optimal threshold for the model's knowledge boundary expression on 100 labeled samples from the TriviaQA training set, aiming to maximize the model's S_{aware} score.

A.2 Prompt-based Methods

Prompt-based methods directly prompt LLMs to declare their knowledge boundaries in textual form, without needing to access the internal signals of

Prompt-based Method	Prompt
Prior Prompt	`Do you know the answer to the following question honestly? If you know, output Yes, otherwise output No, just say one word either Yes or No\n{Q}`
Posterior Prompt	`Are you sure that the answer to the following {Q} is the following {A}? If you are sure, output Sure, otherwise output Unsure, just say one word either Sure or Unsure`
In-context IDK	`Answer the following questions like examples. When you do not know the answer, output Unknow.\nExamples:\nQuestion: Which is the largest island in the Mediterranean Sea?\nAnswer: Sicily\nQuestion: Which country will host the 2016 European Nations football finals?\nAnswer: France\nQuestion: Actress Audrey Hepburn won her only Oscar for which film?\nAnswer: Roman Holiday\nQuestion: Who leads the Catholic Church?\nAnswer: Unknow\n\nYou should only output the answer, without any extra information or explanations. Do not repeat the question. If there are multiple answers, just output the most likely one. The answer should not be a sentence, just a phrase part of the answer. Here is your question: Question: {Q}`
Verbalize Uncertainty	`Provide your best guess and the probability that it is correct (0.0 to 1.0) for the following question. Give ONLY the guess and probability, no other words or explanation. For example:\n\nGuess: <most likely guess, as short as possible; not a complete sentence, just the guess!>\nProbability: <the probability between 0.0 and 1.0 that your guess is correct, without any extra commentary whatsoever; just the probability!>\n\nThe question is:\n{Q}.`

Table 7: Instructional prompts used in the prompt-based method.

the model. Table 7 shows the prompts we used in the prompt-based methods.

A.3 Fine-tuning Methods

We consider two conventional fine-tuning methods as baselines. These fine-tuning methods use the same training set as our approach, but they sample training data based on labels rather than model signals. **Fine-tune** is a conventional instruction fine-tuning method, where the model is fine-tuned directly on question-answer pairs. Regardless of whether the model answers correctly, the fine-tuning target is always the ground truth. **IDK-FT** first lets the model predict the answer to a question. The fine-tuning target depends on whether the model's response matches the ground truth. If it matches, the ground truth is used as the target; if it doesn't, the target is replaced with "Unknow".

B Experimental Settings

B.1 Dataset Details

We use three QA datasets: TriviaQA (Joshi et al., 2017), NQ (Kwiatkowski et al., 2019), and PopQA (Mallen et al., 2023) to construct our test data for evaluating the model's self-awareness. These datasets consist of single-hop factual questions, which do not involve the model's reasoning capabilities, making them more suitable for testing the model's internal knowledge. Additionally, the answers in these datasets are in short-form phrases, avoiding the biases that might be exposed through long-form generation. We use a closed-book, free-form setup to evaluate the models' ability to express their knowledge boundaries. We use exact match to determine whether the model answers correctly or expresses the unknown. The test sets of TriviaQA, NQ, and PopQA contain 11,313, 3,610, and 14,267 samples, respectively. We randomly sampled 2,000 examples from each of the TriviaQA, NQ, and PopQA test sets as our test data. As shown in Figure 8, **the model exhibits different knowledge boundaries across different datasets, and different models also show varying knowledge boundaries on the same dataset.** The proportions of T_k and T_{unk} differ across the test datasets, but the model's self-awareness score

Model	TriviaQA	NQ	PopQA
Llama2-Chat-7B	45.2	16.6	21.7
Llama2-Chat-13B	52.0	21.9	23.5

Table 8: The accuracy of vanilla LLMs on our test data. It represents the portion of knowledge that the model knows and can answer (Known Knows).

Type	Prompt
Prior Awareness	1. `Do you know the answer to the following question honestly? If you know, output Yes, otherwise output No, just say one word either Yes or No\n{Q}`
	2. `Do you have knowledge of the answer to this question? If you do, reply with Yes; if not, reply with No, just say one word either Yes or No\n{Q}`
	3. `Is the answer to this question known to you? Reply Yes if it is; otherwise, reply No, just say one word either Yes or No\n{Q}`
	4. `Do you have enough information to answer the following question? If yes, answer Yes; if no, answer No, just say one word either Yes or No\n{Q}`
Posterior Awareness	1. `Are you certain that ‘A’ is the correct answer to ‘Q’? If certain, answer Sure; if not, answer Unsure, just say one word either Sure or Unsure\nQ: {Q}\nA: {A}\n`
	2. `Do you believe with certainty that ‘A’ is the correct answer to ‘Q’? If yes, answer Sure; if not, answer Unsure, just say one word either Sure or Unsure\nQ: {Q}\nA: {A}\n`
	3. `Are you certain that your answer ‘A’ to ‘Q’ is based on accurate information? If so, answer Sure; if not, answer Unsure, just say one word either Sure or Unsure\nQ: {Q}\nA: {A}\n`
	4. `Do you trust the information that led to your answer ‘A’ to ‘Q’? If confident, answer Sure; if not, answer Unsure, just say one word either Sure or Unsure\nQ: {Q}\nA: {A}\n`

Table 9: Prompts used to test the consistency of knowledge boundary expression under different prompts.

S_{aware} is calculated by averaging the scores corresponding to T_k and T_{unk}, thus not being affected by sample imbalance. Since we use the TriviaQA training set as the training data, the NQ and PopQA datasets, which have distributions different from TriviaQA, serve as out-of-distribution test sets with varying knowledge boundary distributions.

B.2 Prompt for Consistency Evaluation

We used the prompts in Table 9 as the prompt pool for testing the consistency of knowledge boundary expression under different prompts. We utilized GPT-4o to generate different prompts that assess the model's ability to express knowledge boundaries, categorizing them into two types.

B.3 Implementation Details

For our experiment, we choose to use the LLaMA2-Chat (Touvron et al., 2023) model. Based on the pre-trained LLaMA2 model, LLaMA2-Chat is a model that has undergone instruction tuning and RLHF (Stiennon et al., 2020), thereby acquiring the capability to follow instructions. We use the 7B and 13B versions of the LLaMA2-Chat model. We set the thresholds δ_k and δ_{unk} to 0.99 and 0.4, respectively. Due to the large number of instances, we sort the confidence scores from the TriviaQA training set and designate the bottom 10% as D_{unk} and the top 20% as D_k, resulting in approximately 23,000 instances in total. We use LoRA for model fine-tuning, setting r=8, alpha=16, and dropout=0.05. During training, we set the initial learning rate to 1e-4, the final learning rate to 3e-4, the warmup phase to 300 steps, and we train for 700 steps. We conduct all our experiments on 4 NVIDIA A800 80GB GPUs.

C Experimental Supplement

C.1 Effectiveness of Model Signals

We also illustrate the effectiveness of the confidence calculation method through an empirical study. We obtain the model confidence for Llama2-chat-7B on the Trivia-QA training set using three different methods. We divide the model's responses into two parts based on whether the answers are correct and calculate the sample distribution for each part. As shown in Figure 4, there is a significant difference in the confidence distribution

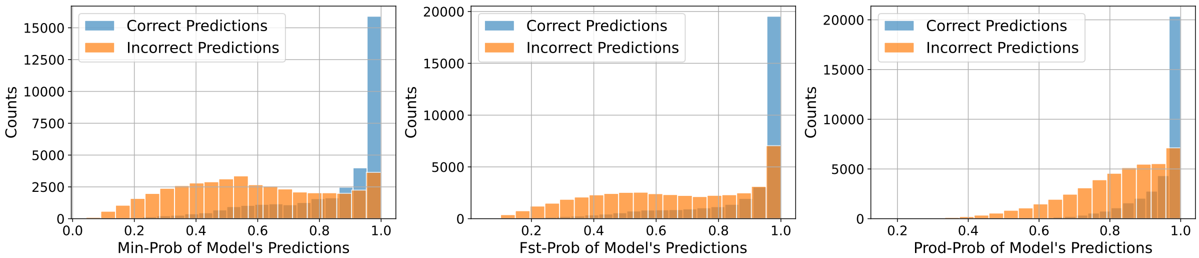

Figure 4: Distribution of model predictions regarding confidence for Llama2-Chat-7B on Trivia-QA. Confidence is calculated using Min-Prob, Fst-Prob, and Prod-Prob from left to right.

between the Correct Predictions and Incorrect Predictions. Predictions with confidence less than 0.4 are mostly incorrect, while the confidence of correct predictions is generally 1.0. This indicates that the model signals can reflect the model's confidence, implying whether the model possesses the corresponding knowledge.