Query-Level Uncertainty in Large Language Models

Lihu Chen[1]**, Gaël Varoquaux**[2]
[1] Imperial College London, UK
[2] Soda, Inria Saclay, France
lihu.chen@imperial.ac.uk
gael.varoquaux@inria.fr

Abstract

It is important for Large Language Models to be aware of the boundary of their knowledge, the mechanism of identifying known and unknown queries. This type of awareness can help models perform adaptive inference, such as invoking RAG, engaging in slow and deep thinking, or adopting the abstention mechanism, which is beneficial to the development of efficient and trustworthy AI. In this work, we propose a method to detect knowledge boundaries via *Query-Level Uncertainty*, which aims to determine if the model is able to address a given query without generating any tokens. To this end, we introduce a novel and training-free method called *Internal Confidence*, which leverages self-evaluations across layers and tokens. Empirical results on both factual QA and mathematical reasoning tasks demonstrate that our internal confidence can outperform several baselines. Furthermore, we showcase that our proposed method can be used for efficient RAG and model cascading, which is able to reduce inference costs while maintaining performance. The code is available at ⌂ https://github.com/tigerchen52/query_level_uncertainty

1 Introduction

Large language Models (LLMs) have their knowledge boundaries (Li et al., 2024; Yin et al., 2024; Ren et al., 2025), which means that there are certain problems that they cannot provide accurate outputs. It is crucial for LLMs to be self-aware of their limitations, i.e., *know what I know and know what I don't know* (Kadavath et al., 2022; Amayuelas et al., 2024).

Possessing awareness of knowledge boundaries provides several advantages in developing efficient and trustworthy AI. First, if LLMs can identify known-unknown or simple-hard queries, they can smartly perform *adaptive inference* to balance the trade-offs between computational cost and output quality.
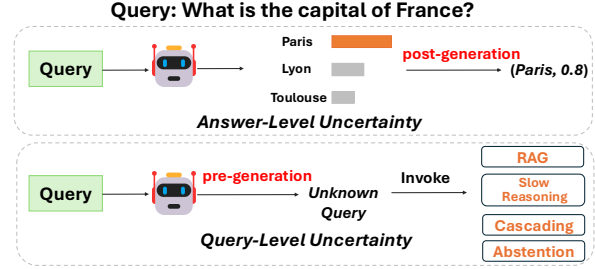

Figure 1: Illustrating the difference between answer-level and query-level uncertainty. Query-level uncertainty estimating known or unknown queries (*knowledge boundary*) before generating answers, which is useful for adaptive inference, e.g., efficient RAG and fast-slow reasoning.

put quality. For queries beyond their parametric knowledge, they can choose to find relevant external knowledge via RAG (Lewis et al., 2020) or tool calls (Schick et al., 2023). When faced with hard problems, LLMs can engage in slow (or deep) thinking to improve their outputs, which is also known as test-time scaling (Snell et al., 2024; Zhang et al., 2025). Alternatively, another solution is to defer a complex problem to a larger model via model cascading (Dohan et al., 2022; Gupta et al., 2024). This adaptive inference ensures that computational resources are allocated effectively, which reduces costs while maintaining performance. Second, estimating whether a query is answerable enhances the honesty and trustworthiness of LLMs. When LLMs identify uncertain queries, they can use the abstention strategy (Wen et al., 2024) to withhold responses, which is important in high-stakes domains like healthcare (Tomani et al., 2024).

In this work, we propose a new concept, *Query-Level Uncertainty*, to estimate a model's knowledge with regard to a given query. The research question here is: *Given a query, can we determine if the model is able to address it without generating*

any tokens? Most existing work focus on answer-level uncertainty, which measures the uncertainty associated with a specific answer, helping us assess the reliability of outputs (Shorinwa et al., 2024; Vashurin et al., 2025). The main distinction here is that we shift from post-generation uncertainty to pre-generation uncertainty, which aims to measure how certain an LLM can solve this query, as shown in Figure 1.

Prior studies propose learning a probe on internal states to predict uncertainties of queries (Gottesman and Geva, 2024; Kossen et al., 2024). Another branch of work attempts to teach LLMs to explicitly express "I don't know" in their responses via fine-tuning methods (Amayuelas et al., 2024; Kapoor et al., 2024; Cohen et al., 2024; Zhang et al., 2024a). One potential issue of these studies is that they often require fine-tuning and training samples, which introduces additional overhead and may limit their generalizability. We aim to introduce a training-free approach to estimate query-level uncertainty, which is simple yet effective.

Our approach relies on self-evaluation across internal layers and tokens, which is called *Internal Confidence*. The proposed approach is based on a simple assumption: LLMs can self-evaluate their knowledge about a query by answering a yes-no question. Inspired by the uncertainty method P(True) (Kadavath et al., 2022), we can compute the probability P(Yes) to indicate the model's confidence. To fully use latent knowledge within LLMs, we compute this kind of P(Yes) at each layer and token position. Following that, we aggregate these signals to obtain the final confidence score. This aggregation is motivated by prior work showing that leveraging logical consistency across layers can improve outputs (Burns et al., 2022; Chuang et al., 2023; Xie et al., 2024). Specifically, we perform a weighted sum across layers and tokens, and the weights are derived from attenuated encoding (Chen et al., 2023), which can control the influence of adjacent units.

To validate the effectiveness of our proposed internal confidence, we conduct experiments on three datasets that cover factual QA and mathematical reasoning tasks. For comparison, we adapt the existing answer-level methods to compute the query-level uncertainty. Experimental results demonstrate that our proposed internal confidence can distinguish known and unknown queries better than various baselines. In terms of applications, we showcase that our proposed method can help efficient RAG and model cascading. On the one hand, internal confidence can guide users to assess the trade-offs between cost and quality when invoking additional services. On the other hand, it brings a "benefit region", where inference overhead can be reduced without compromising performance.

To conclude, we propose a simple yet effective, training-free method to estimate query-level uncertainty, which can determine if a model can address a given query without generating any tokens.

2 Related Work

2.1 Uncertainty Estimation

Existing methods mainly focus on estimating the uncertainty of LLM-generated responses, which aim to provide a score to indicate the reliability of a query-answer pair (Geng et al., 2024; Shorinwa et al., 2024; Vashurin et al., 2025). These approaches often rely on internal states (Chen et al., 2024a) or textual responses (Kuhn et al., 2023), and commonly use calibration techniques to mitigate issues such as overconfidence (Zhang et al., 2024b) and biases (Chen et al., 2024b). Notably, these methods assess post-generation reliability, i.e., they evaluate uncertainty about a particular answer. In contrast, there is limited research on quantifying how well a model can address a query prior to token generation. For example, Gottesman and Geva (2024) propose training a lightweight probe on internal representations to estimate the model's knowledge about specific entities. Similarly, Semantic Entropy Probes (Kossen et al., 2024) suggest that internal model states can implicitly encode semantic uncertainty, even before any output is generated. To the best of our knowledge, this work is the first to formally define query-level uncertainty and investigate it systematically.

2.2 Knowledge Boundary Detection

LLMs should faithfully assess their level of confidence in answering a query. This knowledge boundary awareness (Li et al., 2024; Yin et al., 2024; Wang et al., 2024) is essential to build reliable AI systems, particularly in high-stakes domains such as healthcare and law. A pioneering study by Kadavath et al. (2022) explores whether language models can be trained to predict when they "know" the answer to a given query, introducing the concept of "I Know" (IK) prediction. Based on this idea, subsequent work has proposed methods to help LLMs become explicitly aware of their knowledge

limitations through fine-tuning strategies (Amayuelas et al., 2024; Kapoor et al., 2024). Cohen et al. (2024) further advances this line of research by introducing a special `[IDK]` ("*I don't know*") token into the model's vocabulary, allowing the direct expression of uncertainty in its output. Similarly, R-Tuning (Zhang et al., 2024a) tunes LLMs to refrain from responding to questions beyond their parametric knowledge. While these abstention-based approaches show benefits in mitigating hallucinations (Wen et al., 2024), they often require additional fine-tuning, which introduces overhead and may limit generalizability across models and tasks. In this work, we propose a training-free method to identify the knowledge boundary of an LLM, which offers a more generalizable and efficient alternative to detect the knowledge boundary of LLMs.

3 Preliminary

3.1 Aleatoric and Epistemic Uncertainty

Uncertainty in machine learning is commonly categorized into two main types: aleatoric and epistemic uncertainty (Hora, 1996; Der Kiureghian and Ditlevsen, 2009; Hüllermeier and Waegeman, 2021). These distinctions are often overlooked in the context of LLM uncertainty estimation. Aleatoric uncertainty arises from inherent randomness in the data, such as ambiguous inputs or conflicting annotations. This type of uncertainty is irreducible, as it reflects intrinsic noise in the input data. In contrast, epistemic uncertainty stems from a lack of knowledge, often due to insufficient training data and limited model capacity. Unlike aleatoric uncertainty, epistemic uncertainty is reducible with additional data or advanced modeling. In this work, we focus specifically on epistemic uncertainty, with the goal of evaluating whether an LLM possesses sufficient knowledge to answer a given query. Although it is possible that a dataset may contain some ambiguous queries and noisy labels, we assume that the benchmark datasets used in our experiments are well-curated, and have minimal ambiguity. This assumption allows us to reasonably minimize the impact of aleatoric uncertainty, and study the epistemic uncertainty in a clear way.

3.2 Uncertainty and Confidence

In the context of LLMs, the terms uncertainty and confidence are often used interchangeably (antonyms). However, the two concepts have subtle differences. As noted by Lin et al. (2023), uncertainty is a holistic property of the entire predictive distribution, while confidence refers to the model's estimated confidence level associated with a specific answer. For example, given a query $x =$ "*What is the capital of France*", estimating uncertainty requires the distribution over all possible answers, e.g., *Paris, Toulouse, etc.*, as explained by the semantic entropy framework (Kuhn et al., 2023). In contrast, the conditional probability $P(Y = Paris \mid x)$ can serve as a confidence here to indicate the correctness of a specific answer. In the context of query-level uncertainty, we treat uncertainty and confidence as antonyms, as obtaining full probability distributions over all possible queries for a given model is infeasible.

4 Problem Statement and Method

In this section, we describe our problem definition and introduce our method, *Internal Confidence*, a score that reflects whether an LLM can address a query in its own knowledge, prior to generating tokens.

4.1 Problem Statement

Given a query (including prompt words) $\mathbf{x} = (x_1, \ldots, x_N)$, we aim to quantify the query-level uncertainty, $U(\mathbf{x})$, without generating an answer $\mathbf{y}$. This is different from existing uncertainty methods that estimate the uncertainty associated with a specific generated answer, denoted as $U(\mathbf{x}, \mathbf{y})$. We define that if an LLM can answer a query correctly in greedy decoding, the query falls within the knowledge boundary of the model, and its answer can be reliable. Otherwise, the query falls beyond the model's boundary, and it does not possess sufficient knowledge to answer it. We use this standard to evaluate the estimated query-level uncertainty, i.e., a lower uncertainty indicates a model is more likely to output the correct answer. Although different decoding strategies impact LLM outputs (Song et al., 2024), we aim to measure the internal knowledge of a model in a deterministic way.

Here, we focus on queries with definite answers, which have broad applications such as factual QA and mathematical reasoning. While contentious queries with open answers are also important in areas such as politics and philosophy, they are out of the scope of this work.
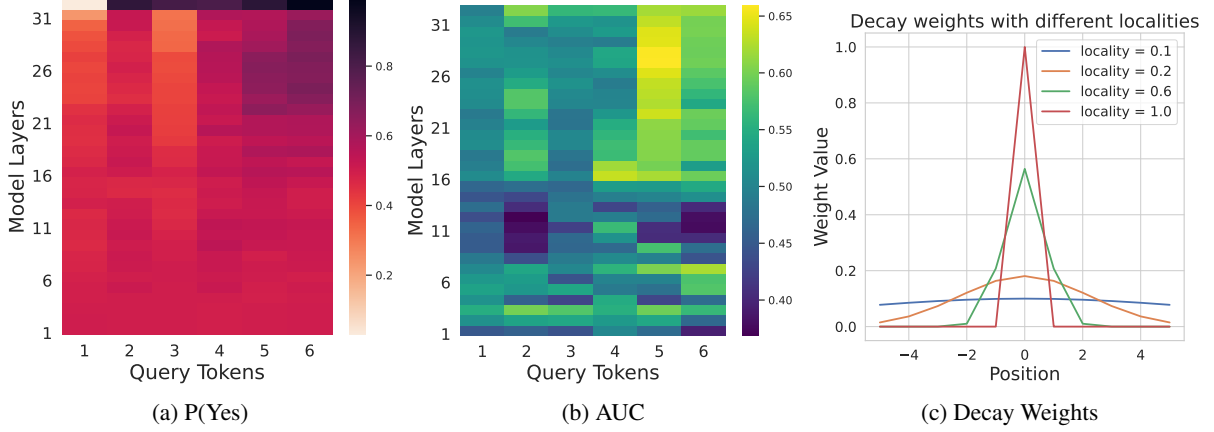
Figure 2: **Left:** the internal P(Yes) across tokens and layers. **Middle:** the AUC of P(Yes) across tokens and layers. **Right:** decay weights with different localities. Model: Llama-8B; Dataset: GSM8K validation set.

4.2 Method

Existing findings reveal that LLMs can express verbalized uncertainty in their responses (Tian et al., 2023; Xiong et al., 2024), which reflects that LLMs can evaluate the answer correctness in their own knowledge. Similarly, we can prompt an LLM to assess its confidence in answering a given query by using a yes-no format: *"Respond only with 'Yes' or 'No' to indicate whether you are capable of answering the {Query} accurately. Answer Yes or No:"*. Following that, we can compute the probability P(Yes) at the last token (x_N):

$$P(\text{Yes}) = \text{Softmax}(\mathbf{W}^{\text{unemb}}_{[\text{Yes, No}]} \cdot \mathbf{h}^{(L)}_N)_{\text{Yes}} \quad (1)$$

where N is the index of the last token in the query, and L is the index of the last layer of the model. $\mathbf{h}^{(L)}_N \in \mathbb{R}^d$ is the hidden state and d is the dimensionality of the hidden representations. $\mathbf{W}^{\text{unemb}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the unembedding matrix that maps the hidden state $\mathbf{h}^{(L)}_N$ to logits over the vocabulary $\mathcal{V}$. P(Yes) can serve as a query-level confidence score here, which is somehow correlated with verbalized uncertainty (Tian et al., 2023), but the main difference is that this method only makes a single forward pass of the query without generating any answer tokens.

However, P(Yes) does fully use internal states of LLMs, which preserves rich latent information about estimating uncertainty (Azaria and Mitchell, 2023; Chen et al., 2024a). Furthermore, prior work demonstrates that using logical consistency across layers can improve outputs (Burns et al., 2022; Chuang et al., 2023; Xie et al., 2024). Therefore, we propose the *Internal Confidence*, which leverages latent knowledge across different layers and

tokens. Let f_θ denote the transformation function for computing hidden states, parameterized by θ. The hidden state for the query x_n of the query at layer l is computed as:

$$\mathbf{h}^{(l)}_n = f_\theta(\mathbf{h}^{(l-1)}_1, \dots, \mathbf{h}^{(l-1)}_n) \quad (2)$$

In total, the model contains $N \times L$ such latent representations, and we can use Equation 4.2 to compute the P(Yes) for each $\mathbf{h}^{(l)}_n$.

Figure 2a shows the average P(Yes) of Llama-8B on the mathematical queries (the validation set of GSM8K (Cobbe et al., 2021)), across layers and query tokens[1]. We observe that the probability increases gradually from low to high layers and from left to right positions, presenting diverse behaviors. If we treat each $P(\text{Yes} \mid \mathbf{h}^{(l)}_n)$ as a confidence score and evaluate Area Under the Curve (AUC), we can obtain an AUC heatmap to show how well the model can distinguish known and unknown queries. As shown in Figure 2b, the top right score is not optimal. Actually, the representation $\mathbf{h}^{(27)}_5$ can achieve the best AUC, and the performance gradually declines in regions surrounding this point. We refer to this optimal point as *Decision Center*. It is important to note that the location of the Decision Center is sensitive to both model architecture and task type.

To improve the naive P(Yes), we can apply a weighted average centering around the decision center, which serves as an ensemble strategy to enhance calibration and expressivity (Zhang et al.,

---

[1]Here, we consider tokens after the {Query}, which means that a model has seen the entire query and is able to guess its knowledge gap.

Method	TriviaQA			SciQ			GSM8K			Avg		
	↑AUC	↑PRR	↓ECE	↑AUC	↑PRR	↓ECE	↑AUC	↑PRR	↓ECE	↑AUC	↑PRR	↓ECE
Phi-3.8B												
Max($-\log p$)	55.5	10.0	-	51.4	2.9	-	55.0	11.3	-	54.0	8.1	-
Predictive Entropy	58.9	17.9	-	51.2	3.9	-	63.6	25.7	-	57.9	15.8	-
Min-K Entropy	59.9	20.0	-	52.7	4.9	-	60.4	17.9	-	57.7	14.3	-
Attentional Entropy	60.6	21.4	-	56.2	9.4	-	52.4	4.4	-	56.4	11.7	-
Perplexity	61.8	24.3	-	57.7	16.6	-	53.6	6.9	-	57.7	15.9	-
Internal Semantic Similarity	48.7	-2.4	0.3	46.9	-5.9	12.2	47.9	-2.6	35.2	47.8	-3.6	15.9
P(Yes)	58.1	16.4	13.9	58.8	16.9	10.8	56.6	12.0	7.6	57.8	15.1	10.8
Internal Confidence (*w/ naive avg*)	58.8	17.3	19.9	52.4	4.5	3.3	54.7	14.7	21.7	55.3	12.2	15.0
Internal Confidence	56.2	13.1	13.9	57.2	15.2	8.2	57.2	12.9	6.0	56.9	13.7	9.4
Llama-8B												
Max($-\log p$)	54.9	11.1	-	51.4	1.9	-	53.3	10.4	-	53.2	7.8	-
Predictive Entropy	58.5	17.7	-	51.4	3.2	-	66.1	28.0	-	58.7	16.3	-
Min-K Entropy	58.1	17.4	-	53.5	7.9	-	57.5	13.2	-	56.4	12.8	-
Attentional Entropy	59.4	18.7	-	57.7	15.2	-	56.1	13.5	-	57.7	15.8	-
Perplexity	58.6	17.1	-	58.3	15.1	-	53.2	4.3	-	56.7	12.2	-
Internal Semantic Similarity	44.1	-14.4	24.4	46.1	-7.1	30.8	52.7	6.7	45.9	47.6	-4.9	33.7
P(Yes)	66.4	33.0	27.5	51.3	2.4	23.7	62.2	24.8	11.6	60.0	20.1	20.9
Internal Confidence (*w/ naive avg*)	67.2	34.4	14.9	58.6	15.4	21.5	59.1	18.7	29.2	61.6	22.8	21.9
Internal Confidence	67.8	34.5	19.1	56.4	13.0	18.9	62.9	27.9	1.3	62.4	25.1	13.1
Qwen-14B												
Max($-\log p$)	56.5	12.4	-	54.1	6.9	-	54.3	13.5	-	55.0	10.9	-
Predictive Entropy	59.3	18.9	-	53.2	6.9	-	66.4	32.6	-	59.6	19.5	-
Min-K Entropy	59.9	20.0	-	55.7	11.3	-	63.0	30.9	-	59.5	20.7	-
Attentional Entropy	59.1	17.2	-	59.4	19.2	-	54.9	3.1	-	57.8	13.2	-
Perplexity	59.1	17.8	-	60.1	20.7	-	54.0	7.3	-	57.7	15.3	-
Internal Semantic Similarity	51.0	2.5	2.0	45.5	-7.7	14.9	47.5	-4.6	33.1	48.0	-3.3	16.7
P(Yes)	63.2	25.8	31.9	61.0	22.4	23.9	54.7	7.5	5.8	59.6	18.6	20.5
Internal Confidence (*w/ naive avg*)	63.3	27.6	8.0	60.5	20.5	15.3	61.7	28.4	36.3	61.8	25.5	19.9
Internal Confidence	69.1	38.4	28.7	65.0	30.8	20.6	62.7	28.4	5.5	65.6	32.5	18.3

Table 1: Overall performances of different query-level uncertainty methods.

2020; Stickland and Murray, 2020). We refer to this process as *Internal Confidence (IC)*, which can be denoted as:

$$\text{IC}(\mathbf{h}) = \sum_{n=1}^{N} \sum_{l=1}^{L} w_n^{(l)} \cdot \text{P}(\text{Yes} \mid \mathbf{h}_n^{(l)}) \quad (3)$$

where $w_n^{(l)}$ is the weight for each $\mathbf{h}_n^{(l)}$. The equation describes a two-step aggregation process. First, we compute a weighted sum across layers for each individual token. Then, we apply a second weighted average over these token-level aggregated scores. Ideally, this process requires a layer weight matrix $\mathbf{W}^{layer} \in \mathbb{R}^{N \times L}$ for the first step and a token weight matrix $\mathbf{W}^{token} \in \mathbb{R}^{1 \times N}$ for the second step. Through this aggregation, we are able to obtain a final confidence score.

In a practical implementation, the decision center is static and fixed to the last token and last layer. However, it is possible to use a hold-out set to identify optimal positions tailored to specific models and tasks. We make this simplification to get rid of the requirement of training samples and aim to obtain better generalizability. Additionally, the layer weight vectors are shared across tokens, which means we need only two weight vectors: $\mathbf{W}^{layer} \in \mathbb{R}^{1 \times L}$ and $\mathbf{W}^{token} \in \mathbb{R}^{1 \times N}$.

To reflect the observations that AUC performances gradually decay from the decision center, we adopt the Attenuated Encoding to compute the above two weight vectors (Chen et al., 2023)

$$\delta_{i,j} = \Phi(d_{i,j}) = \frac{\exp(-w\, d_{i,j}^2)}{\sum_{j=1} \exp(-w\, d_{i,j}^2)} \quad (4)$$

where i is the index of the decision center, $d_{i,j}$ is the relative distance, and $w > 0$ is a scalar parameter that controls the locality value. Locality is a metric that measures how much the weights of a weight vector are gathered in adjacent positions. Given a weight vector for the i-th position $\epsilon_i = \{\epsilon_{i,1}, \epsilon_{i,2}, ..., \epsilon_{i,n}\}$, the locality can be denoted as:

$$\text{Loc}(\epsilon_i) \in [0, 1] = \sum_{j=1} \frac{\epsilon_{i,j}}{2^{|i-j|}} \quad (5)$$

Figure 2c shows the weights computed by Equation 4 with varied localities. This signifies that we can control the influence of neighboring layers and tokens during the averaging process.

Our proposed internal confidence is training-free and efficient, as it requires only a single forward pass of a given query. Since model responses are usually longer than input prompts and invoking

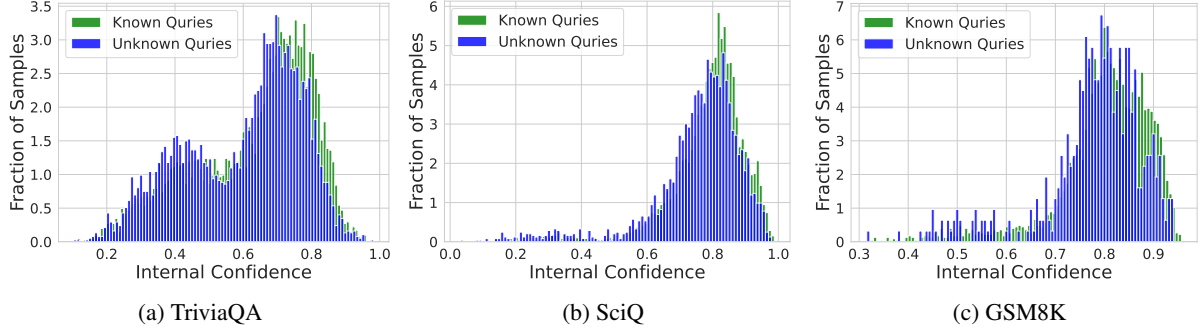|		
(a) TriviaQA	(b) SciQ	(c) GSM8K

Figure 3: We use Internal Confidence of Phi-3.8B to predict whether the corresponding can distinguish known and unknown queries.

external services like RAG adds significant overhead. We hope this pre-generation uncertainty can support adaptive reasoning.

5 Experiments

5.1 Settings

Implementations We provide one positive and one negative example to prompt LLMs, and the target model should follow the examples to output answers. All LLMs use greedy decoding to have deterministic results. The decision center is fixed to the last layer and last token, and we set $w = 1.0$ (Equation 4) for all models and datasets.

Models Three different sizes of LLMs are used in experiments: *Phi-3-mini-4k-instruct* (Abdin et al., 2024), *Llama-3.1-8B-Instruct* (Grattafiori et al., 2024), and *Qwen2.5-14B-Instruct* (Team, 2024). We aim to evaluate if internal confidence can be scaled to different model sizes. Note that internal confidence can be used for models without instruction tuning.

Datasets We evaluate on two factual QA datasets and one mathematical reasoning dataset: TriviaQA (Joshi et al., 2017), SciQ (Welbl et al., 2017), and GSM8K (Cobbe et al., 2021). The first two tasks aim to assess factual knowledge stored in parameters, while GSM8K requires models to self-evaluate their reasoning capabilities. Ground truth of factual QA tasks is a short answer with some entity facts. GSM8k calls for a short answer, but the intermediate reasoning steps have been evaluated as well, following prior work (Kadavath et al., 2022).

We ask a model to generate answers in a greedy decoding way. If the answer is aligned with ground truth, we regard that the model has sufficient knowl-

edge and it falls in its knowledge boundary. For the first two datasets with short answers, we consider an answer to be correct if its Rouge-L (Lin and Och, 2004) of the ground truth is greater than 0.3, which is consistent with prior work (Kuhn et al., 2023). For the GSM8K dataset, we use an LLM evaluator, Mistal-Large (MistralAI, 2024), to assess both reasoning steps and final answer. After that, we can obtain a binary label for each query, which shows if a model is able to address the query.

Baselines We adapt existing answer-level methods to quantify the pre-generation uncertainty, e.g., logit-based uncertainty. Given a query (including prompt words) $\mathbf{x} = (x_1, \ldots, x_N)$, we can obtain a probability for each token $P(x_n \mid x_{<n})$ by performing a forward pass. (1) The baseline $\text{Max}(-\log p)$ measures the query's uncertainty by assessing the least likely token in the query (Manakul et al., 2023). (2) *Predictive Entropy* is defined as the entropy over the entire query tokens (Malinin and Gales, 2021):

$$\text{PE}(\mathbf{x}) = -\sum_{n=1}^{N} \log P(x_n \mid x_{<n}) \qquad (6)$$

(3) *Min-K Entropy* combines the thoughts of the $\text{Max}(-\log p)$ and predictive entropy , which select the top-K of tokens from the query with the minimum token probability (Shi et al., 2024). (4) *Attentional Entropy* is an adapted version of the predictive entropy by performing a weighted sum:

$$\text{AE}(\mathbf{x}) = -\sum_{n=1}^{N} \alpha_n \log P(x_n \mid x_{<n}) \qquad (7)$$

where α_n is the attentional weights for the token x_n. The intuition here is that tokens contribute to the semantic meanings in a different way, and we should
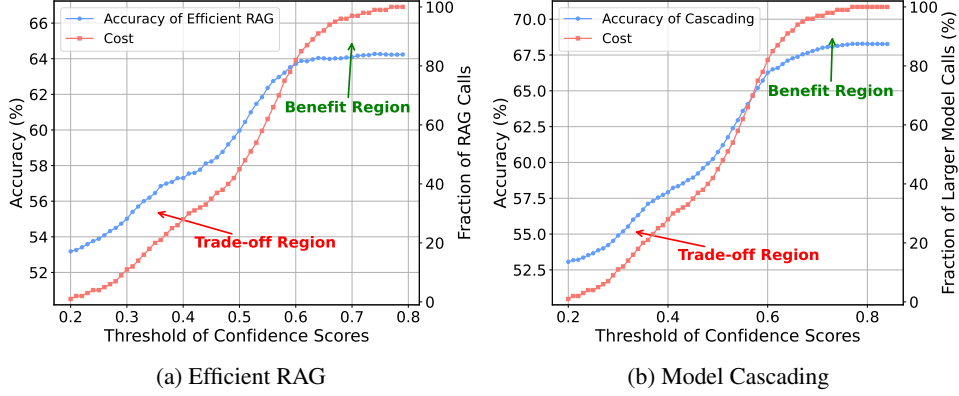
Figure 4: **Left:** We use estimated internal confidence scores to decide whether to invoke RAG. If the internal confidence exceeds a threshold, the model answers the query using its parametric knowledge. Otherwise, it relies on external knowledge for reasoning. The plot shows the accuracy of Phi-3.8B on the TriviaQA dataset under this setting. **Right:** We implement a model cascading seeing with Phi-3.8B (small) and Llama-8B (large) on the TriviaQA dataset. The internal confidence of the smaller model determines whether it answers the query or defers to the larger model when confidence is low.

not treat all tokens equally (Duan et al., 2024). (5) *Perplexity* reflects how uncertain a model is when predicting the next token:

$$PPL = \exp(-\frac{1}{N} \sum logP(x_n \mid x_{<n})) \quad (8)$$

(6) *Internal Semantic Similarity* measures the average similarity among hidden states of different layers $\{\mathbf{h}_N^{(1)}, ..., \mathbf{h}_N^{(L)}\}$, which is inspired by the lexical similarity (Fomicheva et al., 2020). (7) *P(Yes)* is the probability of self-evaluation, which is described in Equation 4.2. (8) *Internal Confidence (w/ naive avg)* is a variant of our proposed internal confidence. The distinction is we apply a naive average to aggregate all scores.

Evaluation Metrics We evaluate uncertainty by assessing whether a method can distinguish *known* and *unknown* queries, which can be treated as ranking problems, i.e., a lower uncertainty means a model is more likely to know the answer to the query. Following prior work (Manakul et al., 2023; Kuhn et al., 2023), we adopt the metrics Area Under the Curve (AUC) and Prediction Rejection Ratio (PRR) (Malinin et al., 2017) to measure this. Additionally, we use the Expected Calibration Error (ECE) to assess the calibration of different methods.

5.2 Internal Confidence Can Identify Known and Unknown Queries

Table 1 shows the overall performances of various query-level uncertainty methods. First, we can

observe that our proposed internal confidence can distinguish known and unknown queries better than other baselines (based on AUC and PRR) on average, especially for larger models such as Llama-8B and Qwen-14B. For example, the average AUC of Qwen-14B is 65.6, which is significantly higher than other baselines. Regarding the calibration (ECE), internal confidence can achieve lower error across models and tasks consistently. These findings indicate the effectiveness of internal confidence. Second, the variant, Internal Confidence (*w/ naive avg*, leads to a decrease in general, which demonstrates that the benefit of using the attenuated encoding to obtain decay weights.

Additionally, Figure 3 shows the how well the internal confidence can distinguish known and unknown queries across three tasks. While the results confirm that our training-free method can predict knowledge boundaries to some extent, there is still considerable room for improvement. We hope this initial effort encourages further research in this direction.

5.3 Internal Confidence Makes LLM Reasoning More Efficiently

Recent studies advance LLM reasoning by introducing additional resources, such as using RAG to obtain external knowledge (Lewis et al., 2020) and inference-time scaling to improve outputs (Snell et al., 2024). However, it is not always necessary to use additional resources, especially for simple queries. Here, we can use our proposed internal
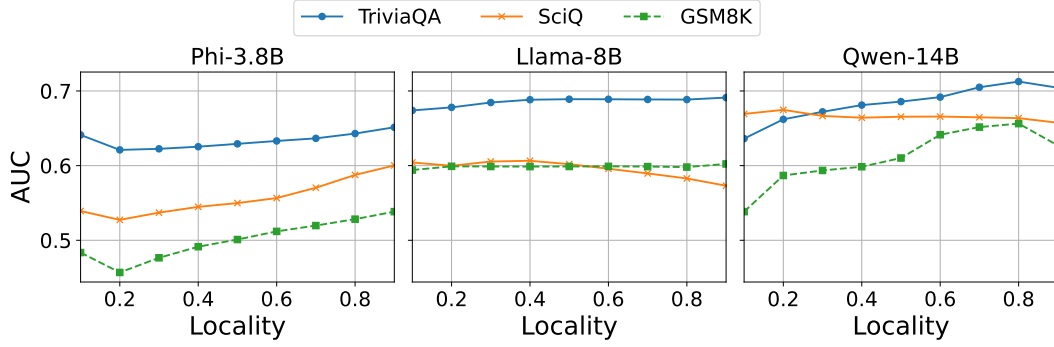
Figure 5: Impacts of locality on validation sets.

confidence to determine when to invoke RAG, slow thinking, or model cascading.

We conduct experiments for two scenarios: (1) *Efficient RAG.* Basically, the internal confidence can serve as a signal of the knowledge gaps of a model. If the score is greater than a threshold, the model is confidence to address the query. Otherwise, it requires the call of RAG. We use the TriviaQA dataset for evaluation. This dataset provides web search results for a query, which can be used as retrieved contexts for RAG. (2) *Model Cascading.* This task aims to achieve cost-performance trade-offs by coordinating small and large models (Dohan et al., 2022; Gupta et al., 2024). Smaller models is responsible for easy missions. If they are aware that the mission is hard to complete, it invokes a larger model. We use a two-model cascade setting with Phi-3.8B and Llama-8B on the TriviaQA dataset. Likewise, if the internal confidence of the smaller model is high, we do not invoke the larger model. Otherwise, the hard query is deferred to the larger model.

Figure 4 shows the results of efficient RAG and model cascading. The trade-off region means that we can carefully select a threshold to control the call of external services, which helps strike a balance between efficiency and performance. The benefit region indicates scenarios where the use of additional resources can be reduced without compromising performance. Results across the two tasks further confirm the effectiveness of Internal Confidence in identifying knowledge gaps. Our method offers practical benefits by reducing inference overhead, which is correlated with computation time and monetary cost.

5.4 Locality Impacts Uncertainty Performance

We introduce attenuated encodings to aggregate probabilities centering around a decision point. The locality of the encoding may impact the performance of estimated uncertainties. To study the influence of the locality, we vary the w in Equation 4 to obtain encoding with different localities and observe how they can impact the estimations. Figure 5 shows the AUC across different datasets and models. We can observe that the locality is correlated with task types and model architecture. For example, Phi-3.8B prefers an extreme locality (1.0) while Qwen-14B has a certain optimal value around 0.8. Regarding different datasets, the influence of locality values displays slightly different behaviors. Although we may need to search an optimal locality for a specific task, we show that an empirical value with ($w = 1.0$, Locality=0.72) can achieve competitive performances across models and datasets.

6 Conclusion

In this work, we propose a new concept called query-level uncertainty, which aims to assess whether a model can address a query without generating any tokens. To this end, we propose the approach, internal confidence, which leverages latent self-evaluation to identify the boundary of a model's knowledge. Experimental results verify the effectiveness of our approach in factual QA and mathematical reasoning. Furthermore, we apply internal confidence to two practical scenarios of adaptive inference, efficient RAG and model cascading. Our findings reveal that our method can identify two regions: a trade-off region and a benefit region. The former means that users can strike a balance between cost and quality by carefully selecting a

threshold of confidence scores. The latter means that users can reduce inference overhead without compromising performance. Although our method can serve as a strong baseline for estimating query-level uncertainty, there is still considerable room for improvement. We hope this study can stimulate future studies in this area.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Alfonso Amayuelas, Kyle Wong, Liangming Pan, Wenhu Chen, and William Yang Wang. 2024. Knowledge of knowledge: Exploring known-unknowns uncertainty with large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6416–6432.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.

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

Lihu Chen, Alexandre Perez-Lebel, Fabian Suchanek, and Gaël Varoquaux. 2024b. Reconfidencing llms from the grouping loss perspective. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1567–1581.

Lihu Chen, Gael Varoquaux, and Fabian Suchanek. 2023. The locality and symmetry of positional encodings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14313–14331.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Roi Cohen, Konstantin Dobler, Eden Biran, and Gerard de Melo. 2024. I don't know: Explicit modeling of uncertainty with an [idk] token. *Advances in Neural Information Processing Systems*, 37:10935–10958.

Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A Saurous, Jascha Sohl-Dickstein, and 1 others. 2022. Language model cascades. *arXiv preprint arXiv:2207.10342*.

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.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.

Daniela Gottesman and Mor Geva. 2024. Estimating knowledge in large language models without generating a single token. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. In *The Twelfth International Conference on Learning Representations*.

Stephen C Hora. 1996. Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. *Reliability Engineering & System Safety*, 54(2-3):217–223.

Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506.

Mandar Joshi, Eunsol Choi, Daniel S 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.

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*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine M Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. 2024. Large language models must be taught to know what they don't know. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*.

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*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Moxin Li, Yong Zhao, Yang Deng, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, and Tat-Seng Chua. 2024. Knowledge boundary of large language models: A survey. *arXiv preprint arXiv:2412.12472*.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (ACL-04)*, pages 605–612.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Andrey Malinin, Anton Ragni, Kate Knill, and Mark Gales. 2017. Incorporating uncertainty into deep learning for spoken language assessment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–50.

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.

MistralAI. 2024. Mistral large: A general-purpose language model. https://mistral.ai/news/mistral-large-2407/.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2025. Investigating the factual knowledge boundary of large language models with retrieval augmentation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3697–3715.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. 2024. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *arXiv preprint arXiv:2412.05563*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2024. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism. *arXiv preprint arXiv:2407.10457*.

Asa Cooper Stickland and Iain Murray. 2020. Diverse ensembles improve calibration. *arXiv preprint arXiv:2007.04206*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Christian Tomani, Kamalika Chaudhuri, Ivan Evtimov, Daniel Cremers, and Mark Ibrahim. 2024. Uncertainty-based abstention in llms improves safety and reduces hallucinations. *arXiv preprint arXiv:2404.10960*.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, and 1 others. 2025. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

Hongru Wang, Boyang Xue, Baohang Zhou, Tianhua Zhang, Cunxiang Wang, Huimin Wang, Guanhua Chen, and Kam-fai Wong. 2024. Self-dc: When to reason and when to act? self divide-and-conquer for compositional unknown questions. *arXiv preprint arXiv:2402.13514*.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. 2024. Know your limits: A survey of abstention in large language models. *arXiv preprint arXiv:2407.18418*.

Zhihui Xie, Jizhou Guo, Tong Yu, and Shuai Li. 2024. Calibrating reasoning in language models with internal consistency. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations*.

Xunjian Yin, Xu Zhang, Jie Ruan, and Xiaojun Wan. 2024. Benchmarking knowledge boundary for large language models: A different perspective on model evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2270–2286.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024a. R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7106–7132.

Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. 2020. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pages 11117–11128. PMLR.

Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. 2024b. Calibrating the confidence of large language models by eliciting fidelity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2959–2979.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, and 1 others. 2025. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*.

A Example Appendix

This is an appendix.