InternalInspector I^2: Robust Confidence Estimation in LLMs through Internal States

Mohammad Beigi[1], Ying Shen[1], Runing Yang[1], Zihao Lin[1], Qifan Wang[2], Ankith Mohan[1], Jianfeng He[1]*, Ming Jin [1], Chang-Tien Lu[1], Lifu Huang[1,3]

[1]Virginia Tech, [2]Meta AI, [3]University of California, Davis

{mohammadbeigi,lifuh}@vt.edu

Abstract

Despite their vast capabilities, Large Language Models (LLMs) often struggle with generating reliable outputs, frequently producing high-confidence inaccuracies known as hallucinations. Addressing this challenge, our research introduces `InternalInspector`, a novel framework designed to enhance confidence estimation in LLMs by leveraging contrastive learning on internal states including attention states, feed-forward states, and activation states of all layers. Unlike existing methods that primarily focus on the final activation state, `InternalInspector` conducts a comprehensive analysis across all internal states of every layer to accurately identify both correct and incorrect prediction processes. By benchmarking `InternalInspector` against existing confidence estimation methods across various natural language understanding and generation tasks, including factual question answering, commonsense reasoning, and reading comprehension, `InternalInspector` achieves significantly higher accuracy in aligning the estimated confidence scores with the correctness of the LLM's predictions and lower calibration error. Furthermore, `InternalInspector` excels at HaluEval, a hallucination detection benchmark, outperforming other internal-based confidence estimation methods in this task.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, from reasoning to question answering (Zhao et al., 2023; Zhou et al., 2023; Wang et al., 2024; Liang et al., 2022). Despite these advancements, LLMs still face significant challenges in hallucinating facts (Ji et al., 2023a; Li et al., 2023a; Ji et al., 2023b; Huang et al., 2023) and providing robust confidence estimates for their predictions (Bommasani et al., 2022; Kuhn et al., 2023; Jiang et al., 2021a)[1]. This results in LLMs delivering confident but incorrect information, undermining their reliability and affecting their potential applications in real-world scenarios. Therefore, a well-established confidence estimator is essential for users to determine when to trust the outputs of LLMs and identify hallucinations in the outputs, thereby enhancing the practicality and trustworthiness of LLMs.

Recent research suggests that LLMs exhibit a degree of self-awareness regarding the truthfulness of the generated statements (Kadavath et al., 2022). Studies have investigated the relationship between the accuracy of LLM outputs and the characteristics of their final activation state (Azaria and Mitchell, 2023a; Burns et al., 2023). Although these findings are promising, they primarily focus on easy True/False factual question-answering tasks. Furthermore, relying solely on the final hidden states offers a limited perspective on the intricate internal dynamics of LLMs. The generation process within LLMs involves a sophisticated interplay of internal modules, including attention mechanisms and Feed-Forward Networks (FFNs), which are critical for shaping the generated responses. Recent studies have demonstrated that these internal modules play a crucial role in encoding and recalling the factual and linguistic knowledge essential for accurate predictions in LLMs (Clark et al., 2019b; Kobayashi et al., 2024; Modarressi et al., 2022; Dar et al., 2023; Ferrando et al., 2022; Modarressi et al., 2023). It has also been shown that hallucinations in LLM outputs primarily originate from these internal modules (Geva et al., 2022a; Li et al., 2023c). These studies further lead us to the question: *can the dynamics of these internal modules potentially indicate the confidence of LLMs in their predictions?*

*This work was done before joining Amazon.

[1]Following (Kadavath et al., 2022; Jiang et al., 2021a), we define *confidence* as the probability of a model prediction being correct, differing from *uncertainty* that quantifies the ambiguity in data or model lack of knowledge (Hu et al., 2023).

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 12847–12865
November 12-16, 2024

In response, we first provide a theoretical foundation that highlights the crucial role of internal states in robust and accurate confidence estimation. We further propose `InternalInspector` (I^2), a simple yet robust confidence estimation method that leverages the internal states of LLMs to assess the truthfulness of generated statements across various tasks. Specifically, `InternalInspector` employs contrastive learning (Khosla et al., 2021) upon an encoder, such as a Convolutional Neural Network (O'shea and Nash, 2015) or a Transformer (Vaswani et al., 2017a), to learn meaningful feature representations from the internal states of an LLM such as `LLaMA-2-7B` (Touvron et al., 2023). A binary classifier is trained simultaneously on top of these feature representations to estimate a confidence score for each LLM prediction based on its correctness, either *correct* or *incorrect*.

We evaluate `InternalInspector` together with various existing confidence estimation methods, including logit-based, self-evaluation, and other internal-based approaches on several natural language understanding and generation tasks, including factual question answering, commonsense reasoning, and reading comprehension. Experimental results demonstrate that `InternalInspector` significantly enhances the alignment between accuracy and confidence scores, achieving up to 20.4% improvement in accuracy and 8.9% in Expected Calibration Error (ECE) across the various evaluation tasks. Furthermore, `InternalInspector` excels at identifying hallucinations in generated outputs, notably existing internal-based confidence estimation methods on the HaluEval benchmark (Li et al., 2023a). We also investigate the importance of different types of internal states in confidence estimation and showcase that attention states are particularly meaningful for tasks that require deep contextual understanding, such as open-book question answering and reading comprehension, while feed-forward states are more crucial for tasks centered on factual information, aligning with the recent research finding that the Feed-Forward Networks (FFNs) within Transformer blocks are functioning as key-value memories to encode and retrieve factual and semantic knowledge (Geva et al., 2021).

In summary, our contributions are as follows:

- We pioneer in establishing a theoretical foundation underscoring the importance of internal states of LLMs in confidence estimation.
- We propose `InternalInspector`, a simple yet effective confidence estimation method that leverages the internal states of LLMs, including the attention states, feed-forward states, and activation states.
- Extensive experiments demonstrate that `InternalInspector` provides robust confidence estimates and significantly outperforms existing confidence estimation methods across various natural language understanding and generation tasks.
- `InternalInspector` is also proven effective in recognizing hallucinations in LLM outputs, achieving significantly better performance than various baselines on HaluEval.

2 Related Work

Confidence Estimation for LLMs We summarize existing confidence estimation methods for LLMs into four categories: **(1) Logit-based methods** (Lin et al., 2022b; Jiang et al., 2021b; Kuhn et al., 2023) utilize output probability distributions or entropy to directly measure confidence. However, they mainly reflect the probability distribution over possible tokens (vocabulary space) (Lin et al., 2022b; Si et al., 2022; Tian et al., 2023). **(2) Consistency-based approaches** (Vazhentsev et al., 2023; Portillo Wightman et al., 2023; Wang et al., 2023; Shi et al., 2022; Manakul et al., 2023; Agrawal et al., 2023) evaluate confidence by measuring the agreement among different model responses, highlighting potential inconsistencies. However, these methods require effective measurement of consistency among responses which is usually challenging (Xiong et al., 2024; Jiang et al., 2021b; Li et al., 2022; Ding et al., 2024; Kuhn et al., 2023; Manakul et al., 2023; Zhang et al., 2023). **(3) Self-evaluation methods** (Kadavath et al., 2022; Manakul et al., 2023; Lin et al., 2024a) enable models to internally assess the correctness of their answers, leveraging their introspective capability. This approach often results in circular reasoning, exacerbating initial errors and leading to overconfident inaccuracies (Ji et al., 2023c; Chen et al., 2023). **(4) Internal-based methods** (Azaria and Mitchell, 2023b; Burns et al., 2022) proposed training a linear classifier on the final activation state of LLMs to examine whether it can differentiate between correct and incorrect answers. `InternalInspector` falls into this category but surpasses existing methods by employing feature learning on the entire spectrum

of the internal mechanism of LLMs to understand the sophisticated non-linear operational process. It generalizes effectively across various datasets and applications, offering robust confidence estimates grounded in comprehensive theoretical analysis.

Probing Probing utilizes linear classifiers as tools to analyze and understand the representations within the intermediate layers of neural networks (Alain and Bengio, 2017). Building upon the foundational concept of probing with linear classifiers, we advances this methodology by utilizing contrastive learning to examine the different internal states of LLMs. This approach allows us to investigate the model's confidence about its predictions, scrutinizing attention, feed-forward, and activation states across all layers, providing a more comprehensive view of how internal representations evolve and interact to influence the overall task performance of the model.

Understanding Internal States in LLMs Studies aimed at understanding the inner workings of transformers indicate that while attention should not be directly equated with explanation (Pruthi et al., 2019; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019), it provides significant insights into the model's operational behavior and helps in error diagnosis and hypothesis development (Park et al., 2019; Voita et al., 2019; Vig, 2019; Hoover et al., 2020; Vashishth et al., 2019). Concurrently, research has shown that Feed-Forward Networks (FFNs) within Transformer blocks, functioning as key-value memories, encode and retrieve factual and semantic knowledge (Geva et al., 2021). Experimental studies have established a direct correlation between modifications in FFN output distributions and subsequent token probabilities, suggesting that the model's output is crafted through cumulative updates from each layer (Geva et al., 2022b). Furthermore, recent study (Li et al., 2023c) has utilized internal states of LLM, aiming to improve the overall truthfulness of LLM generation.

3 Confidence Estimation using Internal Representations

3.1 Background: Transformer Architecture

In this work, we primarily focused on confidence estimation for transformer-based LLMs (Vaswani et al., 2017b), as they have been the predominant architecture backbone of most existing frontier LLMs. Given a sequence of input tokens $x = [x_0, \cdots, x_N]$, a transformer-based language model first encodes the tokens into vectors of input representations $h^0 = [h_0^0, \cdots, h_N^0] \in \mathbb{R}^{N \times d}$ at layer 0. The input representations are then updated through a sequence of L transformer layers, where each layer is composed of a MHSA sublayer followed by a FFN sublayer, interconnected by residual connections that facilitate the flow of information between layers. Formally, the representation of h_i^l of token i at layer l is obtained by:

$$h_i^l = h_i^{l-1} + a_i^l + m_i^l, \tag{1}$$

where a_i^l and m_i^l are the outputs from the l-th MHSA layer and FFN sublayers, respectively.

After the transformation through L layers, the representation at the final layer is projected into the vocabulary space to generate the output sequence y. In this work, we focus on the internal states at the final token across all layers, defined as $\theta = \{h_N^l, a_N^l, m_N^l\}_{l=1}^L$. Here, N represents the position of the last token in the input sequence. We select these internal states because they encapsulate the aggregation of all context information and are directly involved in producing the final predictions, making them particularly relevant for identifying the correctness of LLM's prediction.

3.2 Why Internal Representations for Confidence Estimation?

To analyze the importance of internal states Θ in assessing LLM response correctness, let X and Y be the input and output random variables, respectively, and $K(X)$ the oracle response (derived from expert/world knowledge) as the ground truth for a query X. Given an input-output pair (X, Y), we define a Correctness Indicator $C(Y \mid X)$ as a binary random variable, taking the value 1 if Y is correct given X, and 0 otherwise. Our confidence estimator aims to predict $C(Y \mid X)$. We assume that $C(Y \mid X)$ can be represented by a random function $S(K(X), Y)$, dependent on the oracle answer $K(X)$ and the LLM response Y, i.e., the Correctness Indicator is aligned with the oracle's judgment. Thus, the expected value of this function represents the Correctness Probability:

$$\mathbb{E}[S(K(X), Y)] = P(Y \text{ is correct} \mid X).$$

Let $I(\cdot;\cdot|\cdot)$ and $H(\cdot|\cdot)$ denote the conditional mutual information/entropy, respectively. Assuming

$$H(K(X)|X, Y, S(K(X), Y)) \leq \epsilon,$$

where a small Residual Uncertainty ϵ indicates that Y, combined with the correctness indicator $S(K(X), Y)$, effectively captures most information about the oracle answer $K(X)$.[2]

We further assume

$$I(\Theta; K(X) \mid X) - I(Y; K(X) \mid X) \geq \Delta,$$

where the Internal Knowledge Advantage Δ quantifies the additional information about the oracle answer $K(X)$ encoded in the LLM's internal activation Θ, beyond what is revealed in its output Y. A large Δ implies a richer internal understanding compared to the expressed output. [3]

Mathematically, we establish the key result:

$$I(C(Y \mid X); \Theta | X, Y) \geq \Delta - \epsilon. \quad (2)$$

This implies that when the internal knowledge advantage (Δ) is large and the residual uncertainty (ϵ) is small, the internal states (Θ) provide substantial additional information about the correctness of output (Y) beyond what's contained in the input-output pair (X, Y) alone. A detailed proof is provided in Appendix A. Appendices A.1 and A.2 further explore how internal states influence performance across tasks and quantitatively analyze the impact of internal representation informativeness on confidence estimation, respectively.

3.3 `InternalInspector`

Problem Formulation Given the dataset $\mathcal{D} = \{(x_j, y_j, \theta_j)\}_{j=1}^{M}$, each instance j includes an input text x_j, a generated output y_j, and the internal states $\theta_j = \{h^l_{N,j}, a^l_{N,j}, m^l_{N,j}\}_{l=1}^{L}$ of an LLM when generating the output y_j. Here, N signifies the internal states are extracted at the last token of the input sequence. The internal states θ_j include the activation states $\{h^l_{N,j}\}_{l=1}^{L}$, attention states $\{a^l_{N,j}\}_{l=1}^{L}$, and feed-forward states $\{m^l_{N,j}\}_{l=1}^{L}$ of the LLM across L layers when processing x_j.

To effectively analyze the internal states across all layers, we stack each type of internal state along the layer dimension. For instance, the activation states are constructed as $h^{(1:L)}_{N,j} = [h^1_{N,j}; h^2_{N,j}; \cdots; h^L_{N,j}] \in \mathbb{R}^{L \times d}$ where ; denotes the concatenation along the layer dimension and d is the feature dimension. Similarly, we form the attention states $a^{(1:L)}_{N,j}$ and the feed-forward states $m^{(1:L)}_{N,j}$ both in $\mathbb{R}^{L \times d}$. We further construct the stacked internal states tensor, denoted as $\theta_j = [h^{(1:L)}_{N,j}, a^{(1:L)}_{N,j}, m^{(1:L)}_{N,j}] \in \mathbb{R}^{L \times d \times 3}$ for instance j, capturing the entire internal dynamics of the LLM for instance j.

We formulate the task as learning a function g that takes θ as input and outputs a confidence score c indicating the correctness of y. Each instance (x_j, y_j, θ_j) is associated with a golden binary label c_j based on whether the LLM's prediction y_j is correct, where:

$$c_j = \begin{cases} 1 & \text{if } y_j \text{ is correct} \\ 0 & \text{if } y_j \text{ is incorrect.} \end{cases}$$

Supervised Contrastive Learning `InternalInspector` employs a supervised contrastive learning framework that learns to differentiate the distinctive characteristics associated with correct and incorrect output, relying solely on the internal states. It consists of an encoder, such as a Convolutional Neural Network (CNN) (O'shea and Nash, 2015) or a Transformer (Vaswani et al., 2017b) (architecture detailed at Appendix C), for encoding the stacked internal states $\theta_j \in \mathbb{R}^{L \times d \times 3}$ into a compact representation $z_j = Enc(\theta_j)$. Subsequently, a multilayer perceptron (MLP) classifier is utilized to predict the correctness of the LLM's output y_j via $\hat{c}_j = f(z_j)$. Aligning with the problem formulation, the overall function g, which maps the internal states to the confidence scores, is defined as $\hat{c}_j = g(\theta_j) = f(Enc(\theta_j))$.

`InternalInspector` employs a combination of contrastive loss (Chen et al., 2020) and classification loss to learn fine-grained differences in the internal states that correlate with output correctness. For the contrastive loss, we first organize mini-batches by selecting an anchor embedding z_j from the dataset. For each anchor z_j, we randomly sample one positive embedding z^+_j from the set $Z^+_j = \{z_e \mid c_e = c_j\}$ ensuring that both the anchor and the positive embedding correspond to predictions with the same correctness, i.e., $c_j = c_e$. Additionally, we also sample E negative embeddings $z^-_j \in Z^-_j$, where $Z^-_j = \{z_e \in Z \mid c_e \neq c_j\}$, representing the set of embeddings whose associated predictions y_e

[2]The Residual Uncertainty (ϵ) Tends to be small when the Correctness Indicator is informative and the task is simple.

[3]Empirical evidence supports a large Δ. LLM internal states (Θ) are repositories of open-world knowledge (Geva et al., 2021; Dai et al., 2022; Meng et al., 2022), often containing information not fully expressed in outputs. Even incorrect responses can still possess relevant knowledge internally (Li et al., 2023d). Techniques like enhanced prompting (Wei et al., 2022) and self-evaluation (Kadavath et al., 2022; Saunders et al., 2022; Manakul et al., 2023; Ren et al., 2023; Liu et al., 2023; Lin et al., 2024a) further demonstrate the ability to tap into this latent knowledge to improve accuracy, reinforcing the notion of a substantial Δ.

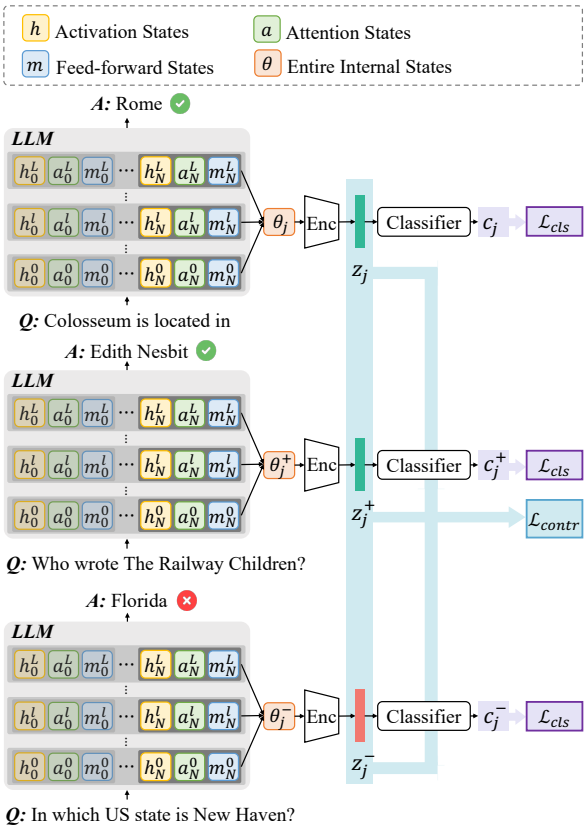

Figure 1: **Overview of our proposed `InternalInspector`.** `InternalInspector` takes in the internal states at the final token across all layers, denoted as $\theta = \{h_N^l, a_N^l, m_N^l\}_{l=1}^L$, as input and outputs a confidence score c indicating the correctness of the LLM's prediction.

differ in correctness from that of the anchor, i.e., $c_e \neq c_j$. Then, the contrastive loss is defined as:

$$\mathcal{L}_{\text{contr}} = -\sum_{j=1}^{M} \log \frac{\exp\left(z_j \cdot z_j^+ / \tau\right)}{\sum_{z_j^- \in Z_j^-} \exp\left(z_j \cdot z_j^- / \tau\right)}, \quad (3)$$

where $\tau \in \mathbb{R}^+$ is a scalar temperature parameter.

For classification, a cross-entropy loss is used to directly optimize the model's ability to classify the embeddings correctly:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{M}\sum_{j=1}^{M} \left(c_j \log(\hat{c}_j) + (1 - c_j)\log(1 - \hat{c}_j)\right), \quad (4)$$

where c_j denotes the golden binary label of the output y_j.

The overall training objective is the combination of the contrastive loss and the classification loss, denoted as $\mathcal{L} = \mathcal{L}_{\text{contr}} + \mathcal{L}_{\text{cls}}$. The combined supervised contrastive loss empowers `InternalInspector` to effectively discern the nuances within the internal states that differentiate correct from incorrect LLM predictions, thus facilitating `InternalInspector` to accurately predict the confidence of the LLM predictions based solely on internal states.

4 Experimental Setting

4.1 Tasks and Datasets

We evaluate `InternalInspector` on one of the most popular autoregressive decoder-only open-source large language models, `LLaMA-2-7B`(Touvron et al., 2023), `Mistral 7B`(Jiang et al., 2023a), and `GPT2-XL`(Radford et al., 2019) on three critical tasks and datasets. For factual closed-book QA, we utilize **TriviaQA** (Joshi et al., 2017) and **MMLU** (Hendrycks et al., 2021). For commonsense reasoning, we employ **CommonsenseQA** (Talmor et al., 2019) and **BoolQA** (Clark et al., 2019a). For reading comprehension, we utilize **SQuAD** (Rajpurkar et al., 2016) and **OpenBookQA** (Mihaylov et al., 2018). Additionally, we also evaluate the capability of `InternalInspector` in detecting hallucinations on **HaluEval** benchmark (Li et al., 2023a).

Due to the page limitation, we present the results for Mistral 7B and GPT2-XL in Appendix D.

4.2 Baselines

To effectively evaluate the effectiveness of `InternalInspector`, we benchmark it against four distinct types of baseline methods:

Logit-Based: Following (Jiang et al., 2021b), the logit-based method utilizes the log probability derived from the output logits as a metric for confidence estimation, under the assumption that higher log probabilities suggest greater confidence.

Self-Evaluation: Following (Kadavath et al., 2022), the Self-Evaluation method initiates a self-assessment phase after generating an answer. After the model generates an answer Y, it feeds both the question X and the generated answer Y back to the model and asks whether the answer is true or false for the question. The confidence is then estimated as the probability of the generated response 'True' $P(\text{True}|X, Y)$.

Temperature Scaling: Following (Desai and Durrett, 2020a), Temperature Scaling adjusts the scale of logits using a scalar hyperparameter T before the softmax operation, modifying the sharpness of the probability distribution.

Last Hidden States: We employ Contrast-Consistent Search (**CSS**) (Burns et al., 2023), which involves training a linear classifier on the final hidden states of statements rephrased in both positive and negative formats, and **SAPLMA** (Azaria and Mitchell, 2023a) that transforms an initial statement into a true/false question and employs a classifier on the final hidden state to map it into the confidence.

4.3 Evaluation Metrics

We assess the performance of confidence estimation using two primary metrics: Accuracy and Expected Calibration Error (ECE) (Guo et al., 2017).

Accuracy This metric measures the proportion of instances where the correctness of the LLM's predictions aligns with the estimated confidence. Specifically, an output of the LLM is considered correct if its estimated confidence score exceeds a predefined threshold and incorrect if it falls below. Following (Burns et al., 2023; Azaria and Mitchell, 2023a; Li et al., 2023a), we set this threshold at 0.5 throughout our experiments, unless stated otherwise.

Expected Calibration Error (ECE) ECE (Guo et al., 2017) quantifies the calibration performance of the models. It is defined as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \quad (5)$$

where n is the total number of samples, M is the number of bins, B_m denotes the m-th bin containing samples with confidences falling within $\left(\frac{m-1}{M}, \frac{m}{M}\right]$, and $|B_m|$ is the number of samples in the m-th bin. Following (Desai and Durrett, 2020b; Kadavath et al., 2022), we use $M = 10$ bins. $\text{acc}(B_m)$ and $\text{conf}(B_m)$ denote the average accuracy and confidence of the samples within B_m, respectively.

5 Results and Discussion

5.1 Main Results

Table 7 shows the performance comparison between `InternalInspector` and baseline confidence estimation methods on various tasks and datasets. As we can see, `InternalInspector` consistently outperforms all baseline methods in terms of accuracy (ACC ↑) and Expected Calibration Error (ECE ↓), demonstrating superior performance across all evaluated tasks and datasets. Specifically, `InternalInspector` achieves significant improvements over the highest-performing baseline,

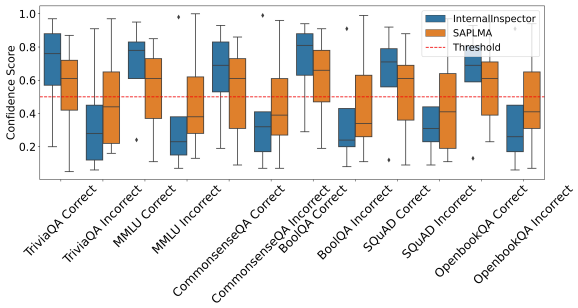

Figure 2: Comparative Distribution of Confidence Scores. Each boxplot indicates the interquartile range of confidence scores. The dashed red line represents the decision threshold at 0.5.

including an average 19.8% increase in accuracy and a 8.95% reduction in ECE for Factual QA. In Commonsense Reasoning, the improvements are 20.4% in accuracy and 6.8% in ECE. For Reading Comprehension, `InternalInspector` enhances accuracy by 18.6 % and lowers ECE by 6.7%.

Additionally, we explore two different architectures for the encoder in `InternalInspector`: a Transformer-based encoder ($\texttt{InternalInspector}_{\text{TF}}$) and a CNN-based encoder ($\texttt{InternalInspector}_{\text{CNN}}$), as presented in Table 7. In general, $\texttt{InternalInspector}_{\text{CNN}}$ outperforms $\texttt{InternalInspector}_{\text{TF}}$ across most datasets. This superior performance is likely to be attributed to CNN's adeptness at capturing the local structure of internal states, thereby providing more effective representations for confidence estimation. The discussions of `InternalInspector` in the following sections are all based on $\texttt{InternalInspector}_{\text{CNN}}$.

Moreover, we conduct an ablation study where $\texttt{InternalInspector}_{\text{CNN}}$ is trained without the contrastive loss. This results in a notable performance decrease across all datasets, underscoring the critical role of contrastive loss in enhancing the model's effectiveness.

Distribution of Confidence Scores We further examine the distribution of the estimated confidence scores from `InternalInspector` compared to those of SAPLMA, the highest-performing baseline. As depicted in Figure 2, there is a clear separation in the confidence score distributions between correct and incorrect predictions across various datasets for `InternalInspector`. `InternalInspector` reliably maintains higher confidence scores for correct answers and lower for incorrect ones, compared to SAPLMA. For `InternalInspector`, both the interquartile ranges and medians for correct answers

Model	Factual QA				Commonsense Reasoning				Reading Comprehension			
	TriviaQA		MMLU		CommonsenseQA		BoolQA		SQuAD		OpenBookQA	
	ACC ↑	**ECE ↓**	**ACC ↑**	**ECE ↓**	**ACC ↑**	**ECE ↓**	**ACC ↑**	**ECE ↓**	**ACC ↑**	**ECE ↓**	**ACC ↑**	**ECE ↓**
Baseline Models												
Logit-Based	0.453	0.202	0.416	0.243	0.576	0.143	0.532	0.254	0.522	0.292	0.464	0.301
Temperature Scaling	0.543	0.181	0.546	0.154	0.667	0.120	0.512	0.212	0.612	0.178	0.594	0.287
Self-Evaluation (3-shot)	0.307	0.465	0.374	0.412	0.312	0.441	0.282	0.590	0.309	0.621	0.368	0.492
CSS	0.552	0.283	0.515	0.245	0.501	0.235	0.568	0.191	0.581	0.243	0.502	0.232
SAPLMA	0.596	0.163	0.606	0.148	0.575	0.123	0.591	0.193	0.617	0.126	0.609	0.157
Our Models												
`InternalInspector`$_{\text{TF}}$	**0.769**	0.081	**0.829**	**0.051**	0.742	0.102	**0.827**	0.099	0.767	0.078	0.723	0.102
w/o Contrastive loss	0.602	/	0.615	/	0.633	/	0.582	/	0.681	/	0.655	/
`InternalInspector`$_{\text{CNN}}$	0.751	**0.073**	0.815	0.054	**0.763**	**0.097**	0.812	**0.083**	**0.807**	**0.051**	**0.791**	**0.098**
w/o Contrastive loss	0.627	0.142	0.641	0.111	0.603	0.168	0.618	0.199	0.660	0.153	0.615	0.171

Table 1: Comparison with baseline confidence estimation methods. Best results are highlighted in **bold**.

Method	QA	Dialogue	Summarization
HaluEval (Llama2-7B)	0.480	0.443	0.476
w/ Knowledge	0.543	0.451	–
w/ CoT	0.427	0.505	0.509
CSS	0.562	0.418	0.481
SAPLMA	0.491	0.459	0.416
`InternalInspector` (Ours)	**0.691**	**0.648**	**0.671**

Table 2: Accuracy (%) of identifying whether a model's output contains hallucinated contents.

Dataset	In-Domain	Intra-Domain	Cross-Domain
Factual QA			
SciQA	0.836	0.737	0.619
MMLU	0.815	0.707	0.598
Commonsense			
BoolQA	0.812	0.694	0.592
CommonsenseQA	0.763	0.673	0.585
Reading Comp.			
SQuAD	0.807	0.683	0.594
OpenBookQA	0.791	0.698	0.539

Table 3: Robustness Across Data Distribution Shifts.

consistently exceed the threshold of 0.5, and remain below this threshold for incorrect predictions. Although SAPLMA similarly positions medians above the threshold for correct responses and below for incorrect ones, it lacks a clear separation across the interquartile ranges, indicating less reliable performance in distinguishing between correct and incorrect predictions. An analysis of high-confidence incorrect answers is in Appendix B.

5.2 Hallucination Detection

To assess the effectiveness of our framework in detecting hallucinations, we apply `InternalInspector` on HaluEval (Li et al., 2023a), a hallucination evaluation benchmark for LLMs. The task involves taking in a question, a corresponding answer, and an optional knowledge context and identifying whether the given answer contains non-factual or hallucinated information.

In our experiments, we employ `LLaMA-2-7B` as the LLM that processes an optional knowledge content, a question, and an answer following the instruction templates in (Li et al., 2023a). The LLM then outputs whether the provided answer is hallucinated or not. We then employ `InternalInspector`, which utilizes the internal states of the LLM, to generate a confidence score indicating the likelihood of the answer being hallucinated. Specifically, we train `InternalInspector` on a 30% subset of the HaluEval and evaluate it on the remaining test split. We apply the same training setup to CSS and SAPLMA, two baseline confidence estimation methods. We also compared with baseline methods from HaluEval, including methods with Chain of Thought (CoT) reasoning (Wei et al., 2022) and knowledge retrieval (Li et al., 2023b), which are zero-shot. As shown in Table 2, `InternalInspector` significantly outperforms the confidence estimation baselines in hallucination detection, suggesting `InternalInspector`'s potential in identifying hallucinations.

5.3 Robustness on Data Distribution Shifts

In this section, we explore `InternalInspector`'s capability to generalize across different datasets, focusing on `Intra-Domain` and `Cross-Domain` settings. In the `Intra-Domain` setting, `InternalInspector` is trained on one dataset of a specific task category and then tested on another dataset from the same task category. For example, within the commonsense reasoning category, the model might be trained on CommonsenseQA and tested on BoolQA. Conversely, in the `Cross-Domain` setting, `InternalInspector` is tested on a dataset of a specific task type while being trained on a combination of datasets from all other

task types that are distinct from the test dataset's category, exemplifying its adaptability across diverse domains. For example, for the `Cross-Domain` scenario involving BoolQA, `InternalInspector` is evaluated on BoolQA while being trained on a combination of datasets from SciQA, MMLU, SQuAD, and OpenBook QA, none of which are within the commonsense reasoning category of BoolQA. Additionally, we include an `In-Domain` setting, where the model is trained and tested on the same dataset to establish a baseline for comparison. Note that, in this experiment, for tasks categorized under FactualQA, we use SciQA (Auer et al., 2023) and MMLU (Hendrycks et al., 2021) due to the similarity in the subject matter they cover, containing question and answers regarding science.

Table 3 showcases `InternalInspector`'s robust performance in the `Intra-Domain` scenarios. Although there is a performance decrement compared to the `In-Domain` setting, `InternalInspector` in the `Intra-Domain` scenario consistently outperforms other baseline methods in the In-Domain one(See Table 7). This strong performance in `Intra-Domain` generalization indicates that the internal states from the same task category exhibit similar patterns. In `Cross-Domain` setting, we observe a larger performance drop, suggesting distinct internal states and patterns across different task categories. This observation aligns with the findings that linguistic and factual knowledge located in different layers of LLMs (Dai et al., 2022; Tenney et al., 2019; Meng et al., 2022; Lin et al., 2024b), resulting in task-specific variations in internal states. Despite these variations, `InternalInspector` still performs comparably to baseline methods that are trained and evaluated on the same dataset, indicating the efficacy of `InternalInspector`.

6 Ablation Study

6.1 Effect of Different Types of Internal States

In this section, we explore the impact of various internal states on `InternalInspector`'s performance, focusing on the role of attention (Attn), feed-forward states (FF), activation state (Act), and their combinations across different tasks. Table 4 demonstrates that when using only one type of internal states, feed-forward states generally prove to be the most influential for confidence estimation, except for the reading comprehension task, where the model using attention states achieves the best performance. This highlights the effectiveness of feed-forward states in tasks that require robust factual recall, while attention states play a crucial role in tasks that necessitate processing and prioritizing extensive text segments for comprehension.

Dataset	Full	FF + Attn	FF + Act	Attn + Act	Attn	FF	Act
Factual QA							
TriviaQA	0.751	0.711	0.724	0.613	0.504	0.703	0.627
MMLU	0.815	0.775	0.767	0.673	0.535	0.717	0.641
Commonsense							
CommonsenseQA	0.763	0.692	0.708	0.619	0.602	0.634	0.617
BoolQA	0.812	0.684	0.739	0.638	0.592	0.693	0.650
Reading Comp.							
SQuAD	0.807	0.771	0.628	0.759	0.728	0.615	0.624
OpenBookQA	0.791	0.719	0.615	0.727	0.707	0.577	0.635

Table 4: Effects of utilizing different combinations of internal states, including attention states (Attn), feed-forward states (FF), and activation states (Act). **Full** represents the use of all types of internal states.

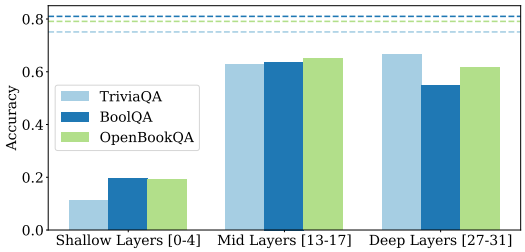

Figure 3: Impact of Internal States from Different Layer Depths.

We also explore the efficacy of various combinations of internal states. The results indicate that integrating multiple types of internal states often yields improvements over using either type of internal states alone. Moreover, models incorporating all types of internal states consistently deliver optimal performance. This suggests that the integration of different types of internal states is necessary to effectively capture the complexities inherent in various tasks, leading to robust confidence estimation methods. These analyses offer insights into how different types or combinations of internal states might influence model performance in various task categories, potentially informing future strategies for the optimal utilization of internal states.

6.2 Impact of Different Layer Depths

In this section, we explore the efficacy of leveraging internal states from different depths of layers within LLMs. Specifically, we analyze the performance of `InternalInspector` across diverse datasets such as TriviaQA, BoolQA, and OpenBookQA, examining how internal states from different layer depths contribute to accurate confidence estimation.

Figure 3 presents the performance of `InternalInspector` when leveraging internal states from shallow (layers 0-4), middle (layers

Size (%)	TriviaQA	MMLU	CommonsenseQA	BoolQA	SquadQA	OpenBookQA
60	0.683	0.721	0.698	0.741	0.688	0.653
70	0.700	0.758	0.705	0.776	0.728	0.702
80	0.716	0.779	0.729	0.798	0.737	0.729
90	0.733	0.811	0.752	0.808	0.803	0.748
100	0.751	0.815	0.763	0.812	0.807	0.791

Table 5: Effects of training size on the performance of $\texttt{InternalInspector}_{\text{CNN}}$ on different datasets

13-17), and deep (layers 27-31) layers. The dashed horizontal line in the figure represents the baseline performance achieved when internal states from all layers are utilized. In general, we observe that the middle layers (13-17) yield the highest performance across different tasks, suggesting that the internal states from the middle layers effectively encode features critical for assessing the correctness of model outputs. Moreover, `InternalInspector` exhibits optimal performance when internal states from all layers are utilized, underscoring the effectiveness of our current model design in leveraging internal states from all layers for confidence estimation.

6.3 Impact of Training Data Size

In this section, we explore the impact of training data size on `InternalInspector` performance. We conducted an ablation study varying the training data size from 60% to 90% of the training split using $\texttt{InternalInspector}_{\text{CNN}}$. As shown in Table 5, our findings indicate that increasing the training data size consistently enhances performance across various tasks.

7 Conclusion

In this work, we propose `InternalInspector`, a simple yet robust confidence estimation method utilizing the internal states of LLMs, including attention, feed-forward, and activation states across all layers. Experimental results underscore `InternalInspector`'s superior performance, which consistently outperforms baseline methods in a variety of natural language processing tasks, including factual question answering, commonsense reasoning, and reading comprehension. Further analysis shows that `InternalInspector`demonstrates strong generalization capabilities within Intra-Domain scenarios. Additionally, `InternalInspector` outperforms other internal-state-based confidence estimation methods in HaluEval, suggesting its potential in hallucination detection.

8 Limitation

`InternalInspector` is specifically designed to leverage the internal states of large language models (LLMs) to estimate the confidence scores of generated responses. Consequently, our proposed model cannot be applied to proprietary LLMs where these internal states are not accessible.

Moreover, in this work, we propose a simple yet effective approach for confidence estimation. While `InternalInspector` demonstrates robust performance across various tasks, we did not extensively explore complex model architectures of `InternalInspector`. Future work could delve into more advanced and complex architectures that might offer improved performance in confidence estimation.

Acknowledgement

This research is partially supported by the award No. 2238940 from the Faculty Early Career Development Program (CAREER) of the National Science Foundation (NSF), the U.S. DARPA ECOLE Program #HR001122S0052, and FoundSci Program #HR00112490370. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

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

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes.

Sören Auer, Dante A. C. Barone, Cassiano Bartz, Eduardo G. Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, Ivan Shilin, Markus Stocker, and Eleni Tsalapati. 2023. The sciqa scientific question answering benchmark for scholarly knowledge. Scientific Reports, 13(1):7240.

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

Amos Azaria and Tom Mitchell. 2023b. The internal state of an llm knows when its lying. Preprint, arXiv:2304.13734.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the opportunities and risks of foundation models. Preprint, arXiv:2108.07258.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. arXiv preprint arXiv:2212.03827.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In The Eleventh International Conference on Learning Representations.

Jiefeng Chen, Jinsung Yoon, Sayna Ebrahimi, Sercan Arik, Tomas Pfister, and Somesh Jha. 2023. Adaptation with self-evaluation to improve selective prediction in LLMs. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 5190–5213, Singapore. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. Boolq: Exploring the surprising difficulty of natural yes/no questions. Preprint, arXiv:1905.10044.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019b. What does bert look at? an analysis of bert's attention. Preprint, arXiv:1906.04341.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. Preprint, arXiv:2104.08696.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16124–16170, Toronto, Canada. Association for Computational Linguistics.

Shrey Desai and Greg Durrett. 2020a. Calibration of pre-trained transformers. CoRR, abs/2003.07892.

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

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using llms: Data perspectives, learning paradigms and challenges. Preprint, arXiv:2403.02990.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8698–8714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022a. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. Preprint, arXiv:2203.14680.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2022b. Self-attention representations reveal the systematicity of semantic knowledge in bert. arXiv preprint arXiv:2203.07442.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. Preprint, arXiv:2012.14913.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, pages 1321–1330. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. Preprint, arXiv:2009.03300.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 187–196, Online. Association for Computational Linguistics.

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ArXiv preprint, abs/2311.05232.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3543–3556. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023a. Survey of hallucination in natural language generation. ACM Computing Surveys, 55(12):1–38.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023b. Survey of hallucination in natural language generation. ACM Comput. Surv., 55(12).

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023c. Towards mitigating LLM hallucination via self reflection. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 1827–1843, Singapore. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. Preprint, arXiv:2310.06825.

Yiding Jiang, Christina Baek, and J Zico Kolter. 2023b. On the joint interaction of models, data, and features. arXiv preprint arXiv:2306.04793.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021a. How can we know when language models know? on the calibration of language models for question answering. Transactions of the Association for Computational Linguistics, 9:962–977.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021b. How can we know when language models know? on the calibration of language models for question answering. Transactions of the Association for Computational Linguistics, 9:962–977.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. arXiv e-prints, arXiv:1705.03551.

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

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised contrastive learning. Preprint, arXiv:2004.11362.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2024. Analyzing feed-forward blocks in transformers through the lens of attention maps. Preprint, arXiv:2302.00456.

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

Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data augmentation approaches in natural language processing: A survey. AI Open, 3:71–90.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. Halueval: A large-scale hallucination evaluation benchmark for large language models. Preprint, arXiv:2305.11747.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jingyuan Wang, Jian-Yun Nie, and Ji-Rong Wen. 2023b. The web can be your oyster for improving large language models. Preprint, arXiv:2305.10998.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023c. Inference-time intervention: Eliciting truthful answers from a language model. ArXiv preprint, abs/2306.03341.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023d. Inference-time intervention: Eliciting truthful answers from a language model. Preprint, arXiv:2306.03341.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. arXiv preprint arXiv:2211.09110.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. Teaching models to express their uncertainty in words. ArXiv preprint, abs/2205.14334.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024a. Generating with confidence: Uncertainty quantification for black-box large language models. Preprint, arXiv:2305.19187.

Zihao Lin, Mohammad Beigi, Hongxuan Li, Yufan Zhou, Yuxiang Zhang, Qifan Wang, Wenpeng Yin, and Lifu Huang. 2024b. Navigating the dual facets: A comprehensive evaluation of sequential memory editing in large language models. Preprint, arXiv:2402.11122.

Genglin Liu, Xingyao Wang, Lifan Yuan, Yangyi Chen, and Hao Peng. 2023. Prudent silence or foolish babble? examining large language models' responses to the unknown. ArXiv preprint, abs/2311.09731.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. Preprint, arXiv:2303.08896.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In Advances in Neural Information Processing Systems.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In EMNLP.

Ali Modarressi, Mohsen Fayyaz, Ehsan Aghazadeh, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2023. DecompX: Explaining transformers decisions by propagating token decomposition. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2649–2664, Toronto, Canada. Association for Computational Linguistics.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 258–271, Seattle, United States. Association for Computational Linguistics.

Keiron O'shea and Ryan Nash. 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Cheonbok Park, Inyoup Na, Yongjang Jo, Sungbok Shin, Jaehyo Yoo, Bum Chul Kwon, Jian Zhao, Hyungjong Noh, Yeonsoo Lee, and Jaegul Choo. 2019. Sanvis: Visual analytics for understanding self-attention networks. Preprint, arXiv:1909.09595.

Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023), pages 326–362, Toronto, Canada. Association for Computational Linguistics.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. 2019. Learning to deceive with attention-based explanations. arXiv preprint arXiv:1909.07913.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Blog, 1(8).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. arXiv preprint arXiv:2206.05802.

Freda Shi, Daniel Fried, Marjan Ghazvininejad, Luke Zettlemoyer, and Sida I. Wang. 2022. Natural language to code translation with execution. Preprint, arXiv:2204.11454.

Chenglei Si, Chen Zhao, Sewon Min, and Jordan Boyd-Graber. 2022. Re-examining calibration: The case of question answering. Preprint, arXiv:2205.12507.

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

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. Preprint, arXiv:2307.09288.

Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. 2019. Attention interpretability across nlp tasks. arXiv preprint arXiv:1909.11218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Artem Vazhentsev, Akim Tsvigun, Roman Vashurin, Sergey Petrakov, Daniil Vasilev, Maxim Panov, Alexander Panchenko, and Artem Shelmanov. 2023. Efficient out-of-domain detection for sequence to sequence models. In Findings of the Association for Computational Linguistics: ACL 2023, pages 1430–1454, Toronto, Canada. Association for Computational Linguistics.

Jesse Vig. 2019. Visualizing attention in transformer-based language models. arXiv preprint arXiv:1904.02679.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. Preprint, arXiv:1905.09418.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A survey on large language model based autonomous agents. Frontiers of Computer Science, 18(6).

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. Preprint, arXiv:2203.11171.

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

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics.

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. Preprint, arXiv:2306.13063.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. arXiv preprint arXiv:2303.18223.

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. arXiv preprint arXiv:2302.09419.

A Detailed Derivation in Section 3.2

First, we have:

$$\begin{aligned}
&I(\Theta; K(X)|X, Y) \\
&= I(\Theta, X, Y; K(X)) - I(X, Y; K(X)) \\
&\geq I(\Theta, X; K(X)) - I(X, Y; K(X)) \\
&= \Delta.
\end{aligned}$$

Next, observe that:

$$\begin{aligned}
&I(S(K(X), Y); \Theta|X, Y) \\
&= I(S(K(X), Y), K(X); \Theta|X, Y) \\
&\qquad - I(K(X); \Theta|X, Y, S(K(X), Y)) \\
&= I(K(X); \Theta|X, Y) \\
&\qquad + I(S(K(X), Y); \Theta|X, Y, K(X)) \\
&\qquad - I(K(X); \Theta|X, Y, S(K(X), Y)) \\
&\geq I(K(X); \Theta|X, Y) \\
&\qquad - I(K(X); \Theta|X, Y, S(K(X), Y)) \\
&= I(K(X); \Theta|X, Y) \\
&\qquad - H(K(X)|X, Y, S(K(X), Y)) \\
&\qquad + H(K(X)|X, Y, S(K(X), Y), \Theta) \\
&\geq I(K(X); \Theta|X, Y) \\
&\qquad - H(K(X)|X, Y, S(K(X), Y)) \\
&\geq I(K(X); \Theta|X, Y) - \epsilon,
\end{aligned}$$

where we used the fact that mutual information is nonnegative. Combining the above inequalities, we obtain the desired result.

A.1 Quantitive assumption

In Sec. 3.2, we demonstrate the correctness C and internal representations Θ are not conditionally independent on (X, Y), $I(C; \Theta|X, Y) > 0$. In this section, we propose a more fine-grained theoretical model for utilizing internal representations, which use a quantified conditional mutual information $I(C; \Theta|X, Y)$ to represent the captured information between Θ and $K(\cdot)$ and introduce the reasoning confidence and knowledging confidence to explain the performance discrepancies among different datasets.

The tested capabilities of LLMs using the QA dataset could be generally divided into two categories: *knowledging and reasoning*. This decomposition is intuitively based on human cognition that knowledging determines whether the LLM is incorporated with the required knowledge to understand the question and answer the question. The reasoning capability determines whether LLM could generate

the correct conclusion. Different QA datasets emphasize different aspects of these capabilities. For instance, mathematical problems focus on the LLM reasoning capability, as they require the LLM to apply logical computation to arrive at the correct answer. Closed-book QA tasks primarily assess LLM knowledging capability. Therefore we assume the confidence signal C generated from the LLM could be decomposed into knowledgeing confidence S_k and reasoning confidence S_r and each component influences the confidence independently. For simplicity, we use a binary version of the confidence score, and the contributions of each component are described in this way:

$$
\begin{aligned}
&P(C=1|S_r=1) = P(C=0|S_r=0) = \alpha \\
&P(C=0|S_r=1) = P(C=1|S_r=0) = 1-\alpha \\
&P(C=1|S_k=1) = P(C=0|S_k=0) = \beta \\
&P(C=0|S_k=1) = P(C=1|S_k=0) = 1-\beta \\
&0.5 \leq \alpha, \beta \leq 1
\end{aligned}
$$

α represents the contribution of reasoning confidence and β represents the contribution of knowledging confidence α and β are greater than 0.5 because two confidence scores contribute positively to the correctness. Then we can derive the probability distribution of C on the joint distribution of S_r and S_k:

$$
\begin{aligned}
&P(C=1|S_r=1,S_k=1) = 0.5(\alpha+\beta) \\
&P(C=1|S_r=1,S_k=0) = 0.5(\alpha+1-\beta) \\
&P(C=1|S_r=0,S_k=1) = 0.5(1-\alpha+\beta) \\
&P(C=1|S_r=0,S_k=0) = 1-0.5(\alpha+\beta) \\
&P(C=0|S_r=1,S_k=1) = 1-0.5(\alpha+\beta) \\
&P(C=0|S_r=1,S_k=0) = 0.5(1-\alpha+\beta) \\
&P(C=0|S_r=0,S_k=1) = 0.5(\alpha+1-\beta) \\
&P(C=0|S_r=0,S_k=0) = 0.5(\alpha+\beta)
\end{aligned}
$$

$I(C;\Theta|X,Y) = H(C|X,Y) - H(C|\Theta,X,Y)$. The first term $H(C|X,Y)$ depends on the data distribution of the selected dataset. To simplify, we assume the dataset D contains the same number of correct and wrong answers. $P(C=1|X,Y\in D) = P(C=0|X,Y\in D) = \frac{1}{2}$. Then $H(C|X,Y) = -2 * (\frac{1}{2})\log(\frac{1}{2}) = \log 2$. The second term $H(C|\Theta,X,Y) = -\sum P(\Theta,X,Y)\sum P(C|\Theta,X,Y)\log P(C|\Theta,X,Y)$, $P(C|\Theta,X,Y) = \sum P(C|S_r,S_k)P(S_r,S_k|\Theta,X,Y)$

Inspired by the interaction tensor (Jiang et al., 2023b), we consider a binarized formulation of the latent features of the internal representations and input-output pair. The three-dimension tensor $\Omega \in \{0,1\}^{M,N,T}$, where the binary label of $\Omega_{mnt} = 1$ indicates the n^{th} data point contains the t^{th} feature and the m^{th} model learns the t^{th} feature. We extend the concept of interaction tensor into our case through two modifications. 1. We do not consider multiple models but care about internal representations from multiple layers. Therefore we use the first axis with size M as the hidden states from each layer. 2. We focus on two specific features of data: reasoning and knowledging. Therefore, for the third axis, we focus on two features t_1 related to reasoning capability and t_2 related to knowledging capability. Then our interaction tensor $\Omega_{mnt} = 1$ indicates the n^{th} question requires the t^{th} capability and the m^{th} hidden states is related to the t^{th} capability. In this project, we leverage all the hidden states to train the confidence estimator, therefore as long as any $m \in M$ hidden state is related to the feature required by the data point, leveraging internal representation is helpful. We define a latent embedding $\Theta_b = \{0,1\}$ of the internal representation Θ and question-answer pair (X,Y). Given the n^{th} data point (X,Y), any $\Omega_{mnt} = 1$ indicates some hidden states capture the required capability, we set $\Theta_b = 1$. Otherwise, $\Theta_b = 0$. We assume Θ_b extracts sufficient information from the question, answer, and internal representation to infer S_r and S_k. And S_r and S_k are conditionally independent on Θ_b. $P(S_r,S_k|\Theta_b) = P(S_r|\Theta_b)P(S_k|\Theta_b)$ Consider the internal representation could provide a binary signal about the reasoning confidence and knowledging confidence and the relations are described in this way:

$$
\begin{aligned}
&P(S_r=1|\Theta_b=1) = P(S_r=0|\Theta_b=0) = \delta \\
&P(S_r=0|\Theta_b=1) = P(S_r=1|\Theta_b=0) = 1-\delta \\
&P(S_k=1|\Theta_b=1) = P(S_k=0|\Theta_b=0) = \epsilon \\
&P(S_k=0|\Theta_b=1) = P(S_k=1|\Theta_b=0) = 1-\epsilon
\end{aligned}
$$

δ and ϵ represent how the internal representation is informative to the model reasoning confidence and knowledging confidence respectively. Then we can enumerate the probability function of C

conditioned on Θ_b.

$$
\begin{aligned}
&P(C=1|\Theta_b=1)\\
=&P(C=1|S_r=1,S_k=1)P(S_r=1,S_k=1|\Theta_b=1)\\
+&P(C=1|S_r=1,S_k=0)P(S_r=1,S_k=0|\Theta_b=1)\\
+&P(C=1|S_r=0,S_k=1)P(S_r=0,S_k=1|\Theta_b=1)\\
+&P(C=1|S_r=0,S_k=0)P(S_r=0,S_k=0|\Theta_b=1)\\
&=\delta\epsilon*0.5(\alpha+\beta)\\
&+\delta(1-\epsilon)*0.5(\alpha+1-\beta)\\
&+(1-\delta)\epsilon*0.5(1-\alpha+\beta)\\
&+(1-\delta)(1-\epsilon)*(1-0.5(\alpha+\beta))
\end{aligned}
$$

$$
\begin{aligned}
&P(C=1|\Theta_b=0)\\
&=P(C=1|S_r=1,S_k=1)P(S_r=1,S_k=1|\Theta_b=0)\\
&+P(C=1|S_r=1,S_k=0)P(S_r=1,S_k=0|\Theta_b=0)\\
&+P(C=1|S_r=0,S_k=1)P(S_r=0,S_k=1|\Theta_b=0)\\
&+P(C=1|S_r=0,S_k=0)P(S_r=0,S_k=0|\Theta_b=0)\\
&=(1-\delta)(1-\epsilon)*0.5(\alpha+\beta)\\
&+(1-\delta)\epsilon*0.5(\alpha+1-\beta)\\
&+\delta(1-\epsilon)*0.5(1-\alpha+\beta)\\
&+\delta\epsilon*(1-0.5(\alpha+\beta))
\end{aligned}
$$

$$
\begin{aligned}
P(C=0|\Theta_b=1)&=1-P(C=1|\Theta_b=1)\\
P(C=0|\Theta_b=0)&=1-P(C=1|\Theta_b=0)
\end{aligned}
$$

The numerical assumption about the probability relation between total confidence, reasoning and knowledging confidence, and internal representation is used to quantitatively analyze the usefulness of leveraging internal representations to predict answer correctness.

A.2 Anslysis in Different Regimes

To simplify the expression, $p_1 = P(C=1|\Theta_b=1)$, $p_0 = P(C=1|\Theta_b=0)$. And $P(\Theta_b=1) = P(\Theta_b=0) = \frac{1}{2}$.

$$
\begin{aligned}
H(C|\Theta_b,X,Y) = &-[p_1\log p_1+(1-p_1)\log(1-p_1)]\\
&-[p_0\log p_0+(1-p_0)\log(1-p_0)]
\end{aligned}
$$

For a fixed LLM, the binarized latent feature Θ_b is a deterministic function of (Θ, X, Y). Therefore $I(C;\Theta_b|X,Y) \le I(C;X,Y,\Theta|X,Y)$.

$$
\begin{aligned}
I(C;X,Y,\Theta|X,Y) &= H(X,Y,\Theta|X,Y)\\
&\quad - H(X,Y,\Theta|X,Y,C)\\
&= H(\Theta|X,Y) - H(\Theta|X,Y,C)\\
&= I(C;\Theta|X,Y)
\end{aligned}
$$

In the following sections, we quantitatively analyze the relationship between the internal representations and confidence through $I(C;\Theta_b|X,Y)$ in different regimes, positive $I(C;\Theta_b|X,Y)$ implies positive $I(C;\Theta|X,Y)$.

A.2.1 Θ_b is highly informative about the S_r and S_k

When the internal representation is highly informative about the S_r and S_k. δ and ϵ could be both close to 1 or both close to 0. The former case indicates that both events often happen together and the latter case indicates one event often happens when the other event does not. Therefore, both cases indicate the strong dependence of C on S_r and S_k. When both δ and ϵ are close to 1, $p_1 \approx 0.5(\alpha+\beta)$ and $p_0 \approx 1-0.5(\alpha+\beta)$. When both δ and ϵ are close to 0, $p_0 \approx 0.5(\alpha+\beta)$ and $p_1 \approx 1-0.5(\alpha+\beta)$. Introduce $\gamma = 0.5(\alpha+\beta) \ge 0.5$ for simplification. In both cases, $H(C|\Theta_b,X,Y) \approx -P(\Theta_b=1)[\gamma\log(\gamma)+(1-\gamma)\log(1-\gamma)] - P(\Theta_b=0)[\gamma\log(\gamma)+(1-\gamma)\log(1-\gamma)] \approx -\gamma\log(\gamma)+(1-\gamma)\log(1-\gamma)$. The conditional entropy is a concave function with respect to γ, which achieves maximum value at $\eta = 0.5$. $I(C;\Theta_b|X,Y) = H(C|X,Y) - H(C|\Theta_b,X,Y) \approx \log 2 - \gamma\log\gamma + (1-\gamma)\log(1-\gamma)$ is non-negative when $\gamma \ge \frac{1}{2}$

A.2.2 Θ_b provides little information about S_r and S_k

When Θ_b provides little information about S_r and S_k, δ and ϵ are close to 0.5. Then $p_1 \approx 0.5$ and $p_0 \approx 0.5$. $H(C|\Theta_b,X,Y) \approx -P(\Theta_b=1)[0.5\log 0.5+0.5\log 0.5] - P(\Theta_b=0)[0.5\log 0.5+0.5\log 0.5] = \log 2$. $I(C;\Theta_b|X,Y) = H(C|X,Y) - H(C|\Theta_b,X,Y) \approx \log 2 - \log 2$ is minimized near to 0.

A.2.3 S_r and S_k contributes to C

When S_r and S_k contributes positively to C, α and β are close to 1. Then we get $p_1 \approx \frac{\delta+\epsilon}{2}$ and $p_0 \approx 1-\frac{\delta+\epsilon}{2}$. Introduce $\eta = \frac{\delta+\epsilon}{2}$ for simplification. $H(C|\Theta_b,X,Y) \approx -P(\Theta_b=1)[\eta\log\eta+(1-\eta)\log(1-\eta)] - P(\Theta_b=0)[(1-\eta)\log(1-\eta)+\eta\log\eta] = -\eta\log\eta+(1-\eta)\log(1-\eta)$. The conditional entropy is a concave function with respect to η, which achieves maximum value at $\eta = 0.5$. $I(C;\Theta_b|X,Y) = H(C|X,Y) - H(C|\Theta_b,X,Y) \approx \log 2 - \eta\log\eta + (1-\eta)\log(1-\eta)$ is non-negative $\eta \ge \frac{1}{2}$

A.2.4 S_r and S_k are not correlated to the C

When S_r and S_k are not correlated to the C, α and β are close to 0.5. Then we get $p_0 \approx 0.5$ and $p_1 \approx 0.5$. $H(C|\Theta_b, X, Y) \approx -P(\Theta_b = 1)[0.5 \log 0.5 + 0.5 \log 0.5] - P(\Theta_b = 0)[0.5 \log 0.5 + 0.5 \log 0.5] = \log 2$. $I(C; \Theta_b|X, Y) = H(C|X, Y) - H(C|\Theta_b, X, Y) \approx \log 2 - \log 2$ is minimized near to 0.

B High-Confidence Incorrect Answers

We further extend our analysis to specifically focus on high-confidence incorrect answers, a critical metric for evaluating the reliability of confidence estimation methods. This analysis is crucial for identifying overconfidence in model predictions, which can have severe implications in high-stakes scenarios. Following the guidelines suggested in (Lin et al., 2022a; Mielke et al., 2022; Lin et al., 2022a), we examine instances where the model, despite incorrect predictions, assigns disproportionately high confidence levels — scores above 0.8. Figure 4 compares the percentage of high-confidence incorrect predictions across various confidence estimation methods. The results demonstrate that `InternalInspector` maintains a significantly lower percentage of high-confidence errors across all datasets and tasks compared to other baselines. This performance underscores the enhanced calibration capability of `InternalInspector`, effectively minimizing the risk associated with overconfident misjudgments and thereby improving the model's overall reliability.

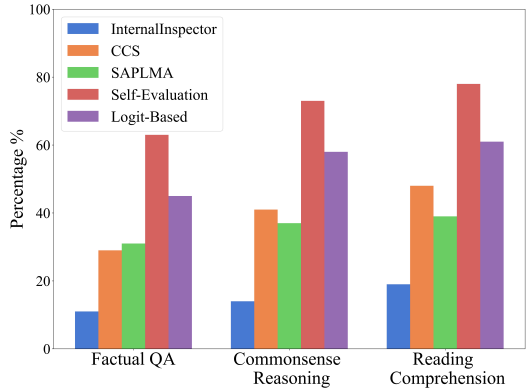

Figure 4: Percentage of high-confidence incorrect answers across various tasks.

C Implementation Detail and Hyperparameters

C.1 Encoder Architecture

We employed the ResNet18 architecture (He et al., 2016) to encode the internal states of the LLM. The deep residual learning framework of ResNet18 efficiently captures relationships both within and across different layers of the LLM. In designing the Transformer as an alternative encoder for our experimental setup, we implemented an 8-layer configuration without an input embedding module, utilizing a model dimensionality of $d = 768$. Each of these layers comprises a self-attention layer and an MLP layer. Atop the feature representations obtained from either the Transformer or the CNN encoder, we train a binary classifier consisting of a multilayer perceptron (MLP) with three layers. This MLP is configured with ReLU activation to effectively process and classify the nuanced, high-dimensional encoded data. To enhance the classifier's robustness and prevent overfitting, L2 regularization and a dropout rate of 0.1 are incorporated into the MLP. It is optimized using a learning rate of 0.001, ensuring stable and efficient learning dynamics.

C.2 Computational Resources

We conduct our experiments on a server equipped with four NVIDIA A100 Tensor Core GPUs. Training `InternalInspector` on various datasets is efficiently completed in under four hours using two of these GPUs.

C.3 Data Split

We train `InternalInspector` on the training split of the datasets and evaluate its performance on the test split.

D Result for Other Models

we have conducted experiments using GPT-2 XL and Mistral 7B, alongside initial tests on LLaMA 7B. Results from GPT-2 XL and Mistral 7B align closely with those from LLaMA, reinforcing our confidence estimation method's robustness and generalizability. This consistency across different architectures and scales underscores the effectiveness of our method, suggesting its applicability to a broader array of LLMs.

Model	Factual QA		Commonsense Reasoning		Reading Comprehension	
	TriviaQA	MMLU	CommonsenseQA	BoolQA	SQuAD	OpenBookQA
Baseline Models						
Logit-Based	46.02	42.82	58.46	64.93	30.12	47.76
Temperature Scaling	50.62	54.09	62.98	67.23	36.83	49.09
Self-Evaluation (3-shot)	31.27	38.12	29.95	42.93	21.42	30.56
CSS	45.10	41.62	48.89	51.50	35.39	40.20
SAPLMA	56.17	55.60	57.20	66.40	41.50	57.10
Our Models						
`InternalInspector`$_{\text{TF}}$	69.82	70.10	69.72	79.10	65.91	69.50
`InternalInspector`$_{\text{CNN}}$	**71.49**	**79.62**	**72.64**	**84.03**	**70.92**	**73.57**
w/o Contrastive loss	66.04	67.63	64.81	76.71	61.29	64.50

Table 6: Main result for Mistral 7B. Comparison with baseline confidence estimation methods. Best results are highlighted in **bold**.

Model	Factual QA		Commonsense Reasoning		Reading Comprehension	
	TriviaQA	MMLU	CommonsenseQA	BoolQA	SQuAD	OpenBookQA
Baseline Models						
Logit-Based	10.72	22.76	18.01	32.63	10.69	28.92
Temperature Scaling	12.29	26.08	20.83	38.49	13.35	32.90
Self-Evaluation (3-shot)	11.92	9.52	10.11	42.50	15.92	35.50
CSS	27.02	21.79	25.20	41.50	26.40	34.10
SAPLMA	33.39	32.89	31.62	46.27	31.45	39.30
Our Models						
`InternalInspector`$_{\text{TF}}$	39.81	39.72	39.09	57.22	38.12	44.40
`InternalInspector`$_{\text{CNN}}$	**42.5**	**41.23**	**44.85**	**52.10**	**40.88**	**47.70**
w/o Contrastive loss	37.11	36.69	36.30	49.80	36.30	45.13

Table 7: Main result for GPT2-XL. Comparison with baseline confidence estimation methods. Best results are highlighted in **bold**.

	TriviaQA	MMLU	Commonsense QA	BoolQA	Squad QA	Openbook QA
Shallow Layer [0-4]	9.10	11.89	8.35	13.43	17.82	7.32
Mid Layer [13-17]	55.23	60.50	57.82	61.90	45.10	58.90
Deep Layer [27-31]	61.82	52.85	64.03	58.48	64.39	46.20
Full [0-31]	71.49	79.62	72.64	84.03	70.92	73.57

Table 8: Analysis of Different Layer on Mistral 7B.

	TriviaQA	MMLU	Commonsense QA	BoolQA	Squad QA	Openbook QA
Shallow Layer [0-4]	4.15	8.62	3.02	4.20	3.40	6.15
Mid Layer [22-26]	21.50	23.02	19.60	35.10	22.80	21.50
Deep Layer [43-47]	34.92	37.30	31.02	43.78	37.50	35.60
Full [0-47]	42.5	41.23	44.85	52.10	40.88	47.70

Table 9: Analysis of Different Layer on GPT2-XL.

Data	In-domain	Intra-domain	Cross-domain
SCIQA	71.70	61.81	48.99
MMLU	79.62	66.20	56.10
BoolQA	84.03	72.52	58.24
Commonsense QA	72.64	61.21	47.44
Squad QA	70.92	59.24	49.18
Openbook QA	73.57	63.42	46.01

Table 10: Analysis of distribution shift on Mistral 7B

Data	In-domain	Intra-domain	Cross-domain
SCIQA	40.52	34.84	20.95
MMLU	41.23	35.54	29.76
BoolQA	52.10	44.11	26.80
Commonsense QA	44.85	38.71	24.22
Squad QA	40.88	33.60	29.03
Openbook QA	47.70	40.36	31.72

Table 11: Analysis of distribution shift on GPT2-XL