PRICE: Preemptive Confidence Estimation in LLMs via Internal Representations

Anonymous ACL submission

Abstract

Large language models have demonstrated strong performance across a wide range of tasks, yet they remain prone to generating incorrect information with high confidence. To mitigate this reliability issue, it is necessary to properly assess model confidence before answer generation. However, existing methods either rely on post-hoc estimation based on generated tokens or sampling results, or require additional training based on supervised learning, which limits their computational efficiency and generalizability. In this paper, we propose **PR**eemptive **I**nternal **C**onfidence **E**stimator (PRICE), which estimates confidence before answer generation by leveraging internal signals of LLMs that can be extracted from a single forward pass. PRICE combines signals that capture pattern agreement across attention heads and the semantic clarity of hidden states. Through experiments on diverse models and benchmarks, we show that PRICE achieves competitive performance against classifier-based methods without requiring additional training. These results suggest that PRICE can contribute to improving both the efficiency of LLMs and their safety in high-risk domains.

1 Introduction

Large Language Models (LLMs) demonstrate exceptional performance across a wide range of natural language processing tasks, including question answering, summarization, and reasoning, and are expanding into diverse application domains. However, despite these performance improvements, LLMs often exhibit an overconfidence problem, generating responses with very high confidence based on incorrect information (OpenAI et al., 2024; Yang et al.; Geng et al., 2024). This can lead to critical errors in domains requiring high accuracy, such as medical and legal fields (Asgari et al., 2025; Dahl et al., 2024). Therefore, techniques for accurately interpreting and quantifying the confidence that models form regarding input queries are not merely performance evaluation metrics, but core components for ensuring safety in determining whether model responses can be trusted and utilized.

Research on model uncertainty and confidence estimation broadly falls into two approaches. First, post-hoc approaches estimate uncertainty from generated outputs, such as output distributions or consistency across multiple sampled responses (Jiang et al., 2021; Fadeeva et al., 2024; Wang et al., 2023; Lin et al., 2024; Manakul et al., 2023a). However, they are only applicable after generation, limiting real time use, and sampling incurs substantial computational overhead due to repeated inference. Second, other work trains separate classifiers or probing models (Azaria and Mitchell, 2023; Binkowski et al., 2025). While often accurate, these methods require additional labeled data and training, and their generalization can degrade under distribution shift to domains different from training data (Zhang et al., 2024).

To overcome these limitations, a method is needed that can estimate confidence before answer generation. Preemptive confidence estimation can enable adaptive computation, in which computational resources are dynamically adjusted according to input difficulty. It can also allow the model to recognize its own uncertainty in advance, thereby preventing overconfidence-induced errors through answer abstention or user warnings. Finally, depending on the situation, it can selectively apply reasoning strategies such as Chain-of-Thought (Wei et al., 2022) or Retrieval-Augmented Generation (Lewis et al., 2020). Therefore, preemptive confidence estimation can be regarded as a key technique for improving the safety, efficiency, and reliability of LLMs.

Accordingly, in this work, we propose **PR**eemptive **I**nternal **C**onfidence **E**stimator (**PRICE**), which estimates confidence before an-

swer generation by leveraging internal signals from transformer (Vaswani et al., 2017)-based LLMs. PRICE uses attention and hidden state information that can be extracted from a single forward pass, jointly measuring the structural consistency of information flow between tokens and the clarity of semantic representations. Specifically, it combines a *consensus signal*, which measures agreement across multi-head attention using Jensen-Shannon Divergence (JSD) (Lin, 2002), with a *salience signal*, which measures semantic clarity based on the coherence and sparsity of hidden states. Through experiments on diverse model architectures and benchmarks, we show that PRICE can effectively estimate uncertainty before answer generation. We also conduct ablation studies to verify the complementarity of the *consensus signal* and the *salience signal*, as well as the validity of our methodological design.

In summary, the contributions of this study are as follows:

- We empirically analyze internal LLM signals before answer generation and show that correct and incorrect predictions exhibit distinguishable patterns in attention disagreement, hidden-state directional consistency, and activation sparsity.
- We propose PRICE, a training-free preemptive confidence estimator that integrates attention-based structural consistency and hidden-state semantic clarity using only a single forward pass.
- Through extensive experiments across diverse models and benchmarks, we demonstrate that PRICE achieves competitive performance against supervised probing approaches without requiring additional classifier training, and provide visual analyses to further interpret the effectiveness of the proposed method.

2 Related Work

Output-based Uncertainty Estimation Methods Early studies treated LLMs as black boxes and estimated confidence based on generated text or token probability distributions. A common approach is to calculate entropy or perplexity using the probabilities of output tokens (Heo et al., 2025; Penny-Dimri et al., 2025; Shorinwa et al., 2025). Low entropy indicates that the model is confident in answer generation. However, token probabilities often do not align with actual accuracy rates due to model calibration errors. To address this, methods have been proposed that perform multiple samplings for the same input and estimate answer consistency (Kuhn et al., 2023; Wang et al., 2023; Manakul et al., 2023b). However, these approaches have the limitation that evaluation is only possible after answer generation, and high computational costs arise from repetitive inference processes (Ma et al., 2025).

Supervised Analysis of Internal Representations Based on the hypothesis that the internal representations of LLMs encode model knowledge and truthfulness, prior studies have investigated the use of attention and hidden states. For example, the entropy of attention heads has been shown to serve as an important indicator for predicting answer correctness, and the properties of hidden states have also been reported to semantically correspond to model interpretations (Zhang et al., 2025; Cencerrado et al., 2025). These internal representations have the advantage that they can be extracted from only a single forward pass. In particular, hidden states are often assumed to directly encode truthfulness or knowledge, and prior work has focused on training probes to detect such properties. Azaria and Mitchell (2023) trained supervised models such as logistic regression on activations from intermediate layers, and Snyder et al. (2024) predicted the likelihood of hallucination using internal signals at the point of first-token generation. In addition, Ji et al. (2024a) showed that model uncertainty can be detected by training probes based on hidden states before answer generation. Although these methods achieve strong performance, they require additional data collection, labeling, and model training costs, and they may suffer from degraded generalization performance, especially in out-of-distribution settings.

3 Preliminary Analysis

Before presenting the proposed method, this section empirically analyzes how the internal information of LLMs varies depending on response confidence. To this end, we compare correct and incorrect samples, focusing on attention patterns and hidden state representations. Each instance was evaluated independently 30 times. For the analysis, we construct 500 instances that generated correct answers in all 30 trials and 500 instances that generated incorrect answers in all 30 trials.

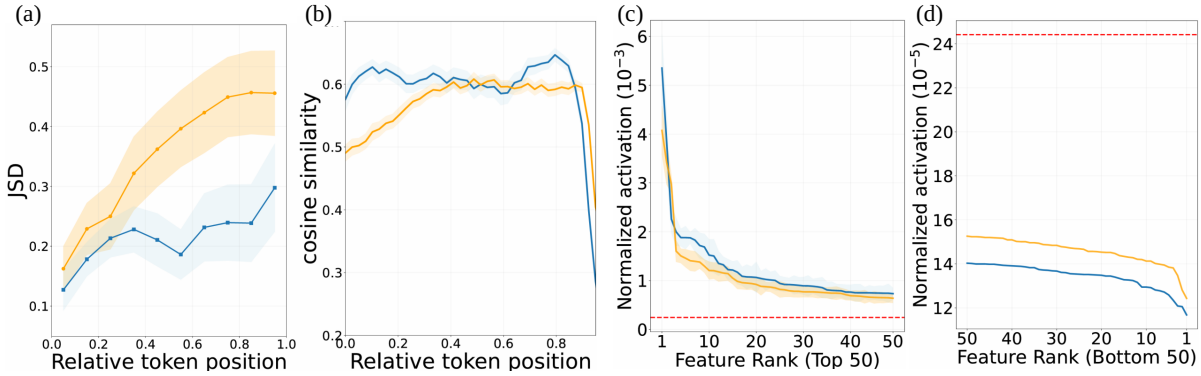

Figure 1: Comparison of internal signal characteristics observed for 500 samples each of consistently correct instances (blue line) and consistently incorrect instances (orange line). (a) Attention pattern dissensus between heads, (b) directional alignment of hidden states within the focus token span, (c) visualization of the 50 features with the largest absolute values in hidden states, (d) visualization of the 50 features with the smallest absolute values in hidden states. Normalized such that the sum of absolute values per feature equals 1, where the red dotted line indicates uniform distribution.

3.1 Extraction of Internal Signals

Attention Let the attention tensor of a transformer based LLM be $\mathbf{A} \in \mathbb{R}^{H \times T \times T}$ for each token position $t \in \{1, \ldots, T\}$ in the input sequence and head $h \in \{1, \ldots, H\}$. At each position t, the mixture distribution averaged over heads is defined as $\mathbf{m}_t = \frac{1}{H} \sum_{h=1}^{H} \mathbf{a}_{h,t}$.[1]

Hidden States According to prior research, due to the autoregressive nature of LLMs, the causal masking structure tends to aggregate information from the entire input in the latter part of the sequence, particularly in the last token (Chen et al., 2024; Azaria and Mitchell, 2023). However, analyzing only the hidden state of a single token risks being sensitive to noise or outliers at that position, so this study comprehensively analyzes the latter region. We denote the hidden state for each token in the input sequence as $\mathbf{h}_t \in \mathbb{R}^d$, and define the focus token span of the last k tokens in the sequence as $I_f = \{T-k+1, \ldots, T\},\ 1 < k < T-1$.

3.2 Consistency of Attention Patterns

Jensen Shannon Divergence To quantify how consistently each head in multi head attention attends to each token of the query, we measure using JSD.

$$\mathrm{JSD}_t = \mathcal{H}(\mathbf{m}_t) - \frac{1}{H} \sum_{h=1}^{H} \mathcal{H}(\mathbf{a}_{h,t}), \qquad (1)$$

[1]Since $\mathbf{a}_{h,t} := \mathbf{A}_h[t,:]$ is a vector that has passed through softmax, each component is non negative and sums to 1, which can be interpreted as a probability distribution.

where $\mathcal{H}(\cdot)$ is the Shannon entropy. A lower JSD_t value indicates more similar patterns.

Observation Figure 1(a) shows the variation of JSD according to token position. Consistently correct instances generally exhibit low JSD, showing a tendency for different heads to attend to similar key tokens. In contrast, consistently incorrect instances show high JSD. This suggests that the internal evidence formation process in the model for consistently incorrect instance inputs is expressed as an unstable pattern across heads.

3.3 Geometry and Sparsity of Hidden States

Directional Alignment of Hidden States To examine how much hidden states within the focus region align in a single direction, we define the mean direction vector as follows: $\bar{\mathbf{h}} = \frac{1}{k} \sum_{t \in I_f} \mathbf{h}_t$. We calculate the cosine similarity between the computed mean direction vector $\bar{\mathbf{h}}$ and each token's hidden state $\mathbf{h}_t$. As can be seen in Figure 1(b), consistently correct instances generally show higher similarity. The sharp drop at the end is interpreted as a boundary effect that occurs as the `eos` token takes on a functional role of concluding the sentence (Clark et al., 2019).

Contrastive Feature Activation To examine the distribution of each feature within hidden states, we normalize the hidden state $\mathbf{h}_t = (h_{t,1}, \cdots, h_{t,d}) \in \mathbb{R}^d$ into an absolute value based probability distribution as in Eq. 2.

$$\mathbf{p}_t = \frac{1}{\|\mathbf{h}_t\|_1} (|h_{t,1}|, \cdots, |h_{t,d}|) \in \mathbb{R}^d, \qquad (2)$$

where $\|\mathbf{h}_t\|_1$ denotes the L1 norm. Figure 1(c) and (d) compare the top 50 and bottom 50 activation values of each feature, respectively. Consistently correct instances show larger values in top features and lower activation values in bottom features. This means they reduce unnecessary noise and emphasize important features more. In contrast, consistently incorrect instances show a weaker tendency in this regard.

3.4 Discussion

In consistently correct instances, multiple heads show distribution patterns that attend to similar tokens, hidden states within the focus region are relatively more aligned with the mean direction, and the activation distribution of each feature within hidden states tends to be sparse. These observations suggest that the internal signals of the model can be structurally related to the confidence of correct answers. In the next section, we design a preemptive confidence measurement method by combining these signals.

4 PRICE

4.1 Overview of PRICE

As illustrated in Figure 2, PRICE integrates two complementary signals: a consensus signal ($S_{\texttt{con}}$) that measures agreement across attention heads, and a salience signal ($S_{\texttt{sal}}$) that measures semantic clarity from hidden states through coherence and sparsity. The overall confidence score is defined as follows:

$$S_{\texttt{PRICE}} = \alpha \cdot S_{\texttt{con}} + (1-\alpha) \cdot S_{\texttt{sal}}, \qquad (3)$$

where $\alpha \in [0, 1]$ controls the relative contribution of the two signals. The two signals capture complementary aspects of model confidence. High attention agreement alone does not necessarily indicate reliable internal representations, while strong feature activation without cross-head agreement may still reflect unstable reasoning. Therefore, PRICE assigns high confidence only when both consensus and salience are jointly satisfied.

4.2 $S_{\texttt{con}}$: Consensus Signal

We calculate the degree of dissensus JSD_t of attention patterns across heads defined in Eq. (1) as an average over all tokens: $\mathrm{JSD}_{\mathrm{all}} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{JSD}_t$. Since the JSD for H distributions lies in the range $0 \leq \mathrm{JSD}_t \leq \log H$,[2] we normalize it by dividing by $\log H$ so that the value ranges between 0 and 1. We then define the consensus signal as in Eq. (4), where values closer to 1 indicate higher agreement:

$$S_{\texttt{con}} = 1 - \frac{\mathrm{JSD}_{\mathrm{all}}}{\log H}. \qquad (4)$$

This definition is not simply a heuristic of similarity/distance, but is precisely interpreted as the mutual information between head identity and observed key tokens. We prove this through Proposition 2 in Appendix D. Additionally, while Eq. (4) treats all heads equally, a method for weighting heads individually is described in Appendix E.

4.3 $S_{\texttt{sal}}$: Salience Signal

$S_{\texttt{con}}$ measures agreement across attention heads, but does not directly guarantee the informativeness of attention itself (Jain and Wallace, 2019). For example, if all heads output attention distributions close to uniform, $\mathrm{JSD}_t \approx 0$ can still produce a high $S_{\texttt{con}}$ value, despite the absence of informative evidence. To complement this limitation, we aim to quantify and use whether sparse patterns are shown in the most important segment for correct answer generation from hidden states. Therefore, in this study, we interpret the consensus signal as an internal instability signal, but complement such cases by subsequently combining it with the salience signal.

Coherence The normalized directional coherence within the focus token span is as follows:

$$C_{I_f} = \frac{\|\sum_{t \in I_f} \mathbf{h}_t\|_2}{\sum_{t \in I_f} \|\mathbf{h}_t\|_2}.$$

If vectors point in similar directions, the activation value for each feature of the sum vector will be large, and if they point in different directions, it will be small. Therefore, the value of C_{I_f} is proportional to the degree of directional alignment of the vectors. Using the formula for C_{I_f}, the norm of the mean vector of hidden states can be calculated as follows:

$$\|\overline{\mathbf{h}}\|_2 = \|\sum_{t \in I_f} \mathbf{h}_t\|_2 = C_{I_f} \cdot \frac{1}{k} \sum_{t \in I_f} \|\mathbf{h}_t\|_2.$$

That is, $\|\overline{\mathbf{h}}\|_2$ is a signal that reflects both the degree of alignment in one direction (C_{I_f}) and the average activation ($\frac{1}{k}\sum_{t \in I_f} \|\mathbf{h}_t\|_2$) of token hidden states within the focus token span. Based on this, we use it as the final coherence measurement as:

$$\mathrm{Coherence} = \phi(\|\overline{\mathbf{h}}\|_2 / \tau),$$

[2] See Proposition 1 in Appendix D.

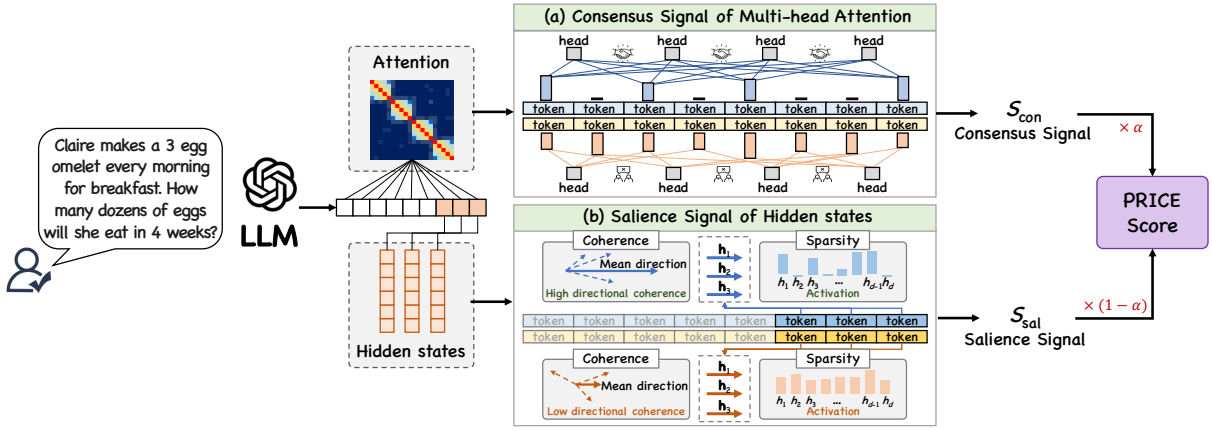

Figure 2: Overview of the PRICE framework for estimating preemptive confidence. (a) Consensus signal is calculated by measuring agreement across heads in multi head attention. Consistently correct instances focus on similar tokens across heads. In contrast, consistently incorrect instances tend to attend to disparate tokens. (b) Salience signal is calculated by measuring coherence and sparsity from hidden states in the focus token span. Consistently correct instances show high directional alignment and sparse activation values. Conversely, consistently incorrect instances show low directional alignment and lack sparse patterns. The final confidence is estimated by a weighted sum of the two signals.

where the monotonic increasing function $\phi : \mathbb{R} \to [0, 1]$ is a function that normalizes coherence.

Sparsity To measure sparsity, we compute the deviation from the uniform distribution $(\frac{1}{d}, \cdots, \frac{1}{d}) \in \mathbb{R}^d$ using the probability distribution $\mathbf{p}_t = (p_{t,1}, \cdots, p_{t,d})$ defined in Eq. (2):

$$G(\mathbf{p}_t) = \frac{1}{2(1-\frac{1}{d})} \sum_{j=1}^{d} \left| p_{t,j} - \frac{1}{d} \right|, \qquad (5)$$

where $G(\mathbf{p}_t)$ is inspired by the Gini coefficient (Hurley and Rickard, 2009). The factor $2(1-\frac{1}{d})$ normalizes the value to have a maximum of 1, which is achieved when $\mathbf{p}_t$ is maximally sparse, i.e., one dimension is 1 and all others are 0. We compute Eq. (5) for all hidden states within the focus token span, and use the average value as the final sparsity measurement.

Salience Signal Using the previously measured indicators, we define and use the salience signal as follows:

$$S_{\texttt{sal}} = \text{Coherence} \times \text{Sparsity}.$$

This is combined in multiplicative form to give a high score only when both coherence and sparsity conditions are simultaneously strong.

5 Experimental Settings

5.1 Models

We used three LLM models with different structures and training characteristics. Specifically, we selected `Mistral-7B-v0.3-Instruct` (Jiang et al., 2023), `Qwen2.5-7B-Instruct` (Qwen et al., 2025), and `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024). All experiments were conducted on eight NVIDIA RTX3090 GPUs.

5.2 Datasets

We used five publicly available benchmark datasets spanning diverse domains and difficulty levels: `MMLU` (Hendrycks et al., 2021), Big Bench Hard (`BBH`) (Suzgun et al., 2022), `GPQA` (Rein et al., 2023), ARC-Challenge (`ARC-C`) (Clark et al., 2018), and `GSM8K` (Cobbe et al., 2021). Detailed statistics for each datasets are provided in Appendix B.

5.3 Baselines

All methods estimate confidence using only the input problem without answer generation. We compare four training-free baselines and two supervised baselines. For training-free baselines, we use: (i) a `prompt`-based self-reported confidence baseline (averaged over three runs), (ii) a `perplexity`-based baseline computed from the conditional likelihood of the problem text, and internal-representation baselines, (iii) `Hidden Score` and (iv) `Attention Score` (Sriramanan et al., 2024). `Hidden Score` summarizes hidden-state activation and distribution characteristics, while `Attention Score` measures the stability of attention patterns across heads. For supervised baselines, we include two probes. (v) `Probe1`: we train a multi-layer

perceptron (MLP) based classifier using the hidden state of the final input token as input, and use the classifier's predicted probability of correctness as the confidence score (Ji et al., 2024b). (vi) `Probe2`: We train a linear probe that takes as input a mean-pooled representation of the input hidden states (Liu et al., 2024). Detailed settings for the baselines are provided in Appendix A.

5.4 Evaluation Metrics

We evaluate confidence estimation performance using smooth expected calibration error (smECE) (Blasiok and Nakkiran, 2024) and the Brier score (Glenn et al., 1950). smECE measures the discrepancy between predicted confidence and conditional correctness rates using kernel smoothing instead of discrete binning. Unlike conventional ECE (Guo et al., 2017), which is sensitive to the number of bins, smECE provides a smoother and more stable estimation of calibration error. While smECE evaluates the overall alignment between confidence and accuracy, the Brier score measures confidence quality at the individual sample level by jointly penalizing overconfident incorrect predictions and underconfident correct predictions.

5.5 Implementation Details

Both attention and hidden states are extracted from the last layer of the LLM. For the salience signal, we apply the $\tanh$ function for normalization, set $\tau = 30$, and define the focus token span as the last 30% of the input sequence.

Weight of `PRICE` `PRICE` combines $S_{\texttt{con}}$ and $S_{\texttt{sal}}$ through a weighted sum parameterized by α. Rather than performing hyperparameter tuning, we estimate α by minimizing the Brier score on a training split. Specifically, given n samples, α^* is obtained by solving:

$$\alpha^* = \arg\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} (S_{\texttt{PRICE}} - y_i)^2.$$

This optimization admits the following closed-form solution:

$$\alpha^* = \frac{\sum_{i=1}^{n} (S_{\texttt{con},i} - S_{\texttt{sal},i})(y_i - S_{\texttt{sal},i})}{\sum_{i=1}^{n} (S_{\texttt{con},i} - S_{\texttt{sal},i})^2}. \quad (6)$$

See Proposition 4 in Appendix D for the derivation. The resulting α^* is the finite-sample counterpart of the population-optimal solution. In addition to minimizing the Brier score, it also helps minimize an upper bound on smECE.[3]

For fairness, we compare `PRICE` in two different settings. When comparing against training-free baselines such as `Prompt`, `Perplexity`, `Hidden Score`, and `Attention Score`, we fix α in `PRICE` to 0.5. In contrast, when comparing against probe-based baselines, we use α^* computed by Eq. (6). To estimate the distributions of $S_{\texttt{con}}$ and $S_{\texttt{sal}}$, we split each dataset into 80% training and 20% test sets, compute α^* on the training set, and evaluate `PRICE` with the resulting α^* on the test set.

6 Experimental Results

6.1 Overall Performance

Table 1 shows the confidence estimation performance on three LLMs and five benchmark datasets, where the weight of `PRICE` is fixed to 0.5. Overall, `PRICE` achieves the best performance in most settings, consistently obtaining lower smECE and Brier scores than existing methods. These results suggest that `PRICE` effectively captures model uncertainty before answer generation. `Prompt` based self-reported confidence generally exhibits high smECE and Brier scores, supporting prior findings (Tian et al., 2023; Groot and Valdenegro Toro, 2024) that LLMs tend to be overconfident when expressing confidence in natural language. The `perplexity` based method shows moderate performance on `ARC-C`, but its performance degrades substantially on more challenging datasets such as `GPQA`. `Hidden Score` and `Attention Score`, which rely on single internal signals, outperform `prompt`- and `perplexity`-based methods in several cases, but show relatively high variance across models and datasets. In conclusion, the results indicate that the two complementary signals in `PRICE` capture distinct aspects of uncertainty and jointly improve confidence estimation performance beyond single-signal approaches.

6.2 Comparison with Supervised Probes

In this section, we compare the performance of `PRICE` using α^* computed by Eq. (6) with that of training-based probes. As shown in Figure 3, training-based probes generally achieve strong performance because they directly learn decision boundaries from labeled representations. In contrast, `PRICE` maintains competitive performance

[3] See Proposition 3 and Lemma 1 in Appendix D.

Model	Method	MMLU		BBH		GPQA		ARC-C		GSM8K	
		smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓
Mistral7B	Prompt	0.3689	0.3255	0.3088	0.3357	0.4111	0.3988	0.2989	0.3041	0.3201	0.3556
	Perplexity	0.2655	0.3303	0.5579	0.5520	0.5303	0.5052	**0.0846**	**0.1796**	0.4115	0.5163
	Hidden Score	0.1001	0.2478	0.1028	0.2389	0.3460	0.3547	0.0988	0.1829	0.1315	0.2972
	Attention Score	0.1017	0.2608	0.1829	0.2720	0.4018	0.3959	0.1767	0.2097	0.0764	0.2511
	PRICE ($\alpha = 0.5$)	**0.0371**	**0.1999**	**0.0234**	**0.1838**	**0.1139**	**0.2134**	0.1195	0.1819	**0.0261**	**0.2020**
Qwen7B	Prompt	0.1958	0.2296	0.1913	0.2196	0.3628	0.3508	0.2115	0.2443	0.4189	0.4336
	Perplexity	0.1784	0.2434	0.4717	0.4722	0.5315	0.5111	0.1259	**0.1114**	0.4255	0.5092
	Hidden Score	0.0864	0.2165	0.0875	0.2523	0.3122	0.3391	0.2226	0.1510	0.1515	0.2709
	Attention Score	0.0983	0.2126	0.0542	0.2452	0.2674	0.3114	0.2293	0.1571	0.1043	0.2161
	PRICE ($\alpha = 0.5$)	**0.0531**	**0.1653**	**0.0316**	**0.1926**	**0.0946**	**0.2216**	**0.0991**	0.1917	**0.0755**	**0.1888**
Llama8B	Prompt	0.1798	0.1969	0.2548	0.3099	0.4518	0.4885	0.1295	0.1573	0.4259	0.3933
	Perplexity	0.2272	0.2914	0.5211	0.5199	0.5620	0.5310	0.1329	0.1665	0.2902	0.3119
	Hidden Score	0.0610	0.2256	0.1658	0.2642	0.3719	0.3746	0.0887	0.1574	0.1077	0.2987
	Attention Score	0.0792	0.2405	0.1743	0.2697	0.2819	0.3248	0.0731	0.1558	0.0823	0.2621
	PRICE ($\alpha = 0.5$)	**0.0257**	**0.1784**	**0.0339**	**0.1941**	**0.1272**	**0.2073**	**0.0428**	**0.1276**	**0.0538**	**0.2110**

Table 1: Confidence estimation performance of PRICE with a fixed weight of 0.5 and training-free baselines across three LLMs and five datasets. Lower smECE and Brier score indicate better performance. The best performance is marked in **bold**, and the second-best performance is underlined. PRICE achieves the best performance in most cases.

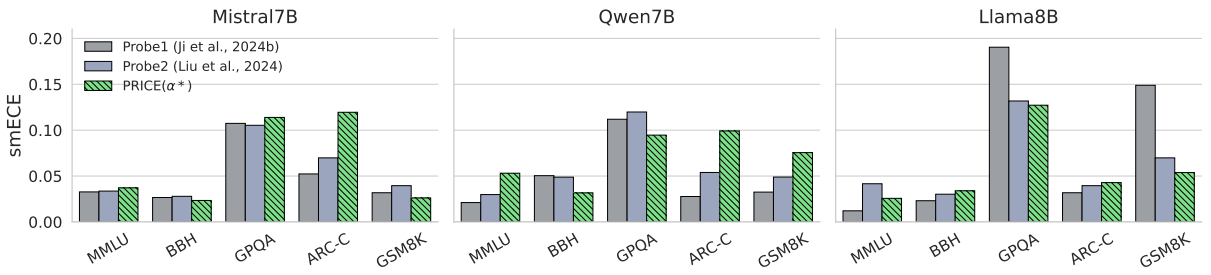

Figure 3: Confidence estimation performance of PRICE with α^* and training-based probes across three LLMs and five datasets. Although probes generally show strong performance, PRICE also achieves competitive performance without requiring separate classifier training. The full results are reported in Table 7 in the Appendix.

against supervised probes without requiring separate classifier training, and even outperforms them in some settings. This suggests that the internal representations of LLMs contain sufficiently reliable confidence information even without additional supervised learning. We further confirm that estimating α^* does not require a large amount of data. Empirically, stable performance can be obtained using only 20% of the full dataset. Additional results are presented in Figure 8 in Appendix C.

7 Ablation Studies

7.1 Complementarity of $S_{\texttt{con}}$ and $S_{\texttt{sal}}$

Figure 4 shows a binned accuracy heatmap analyzing the relationship between $S_{\texttt{con}}$ and $S_{\texttt{sal}}$. We divide the samples into 30 bins and compute the mean correct rate within each bin. The Pearson correlation coefficient between the two signals is 0.13, suggesting that they measure confidence from relatively independent perspectives and provide complementary information. In particular, many bins with low correct rates are observed in the fourth quadrant (high $S_{\texttt{con}}$, low $S_{\texttt{sal}}$), whereas bins with relatively high correct rates appear in the second quadrant (low $S_{\texttt{con}}$, high $S_{\texttt{sal}}$). This asymmetric pattern indicates that relying on a single signal alone is insufficient for stable confidence estimation.

7.2 Effect of the Extraction Layer

Each layer of a Transformer is known to encode different levels of information (Jawahar et al., 2019). Accordingly, we compare confidence estimation performance by extracting attention and hidden states from intermediate layers to the final layer. Figure 5 shows the results on the BBH dataset. Across all three models, smECE tends to decrease

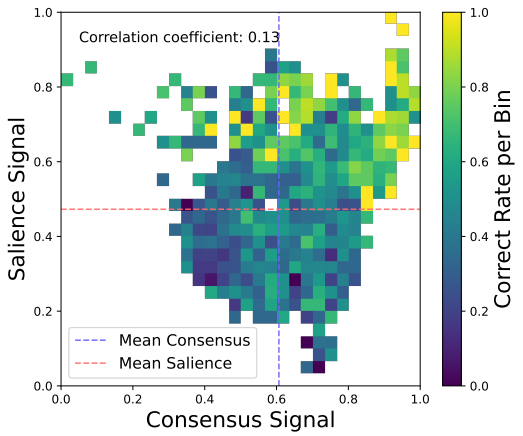

Figure 4: Binned heatmap of the consensus signal and salience signal. Each cell represents the mean correct rate of samples within a 30×30 bin. Cells containing fewer than five samples are not displayed.

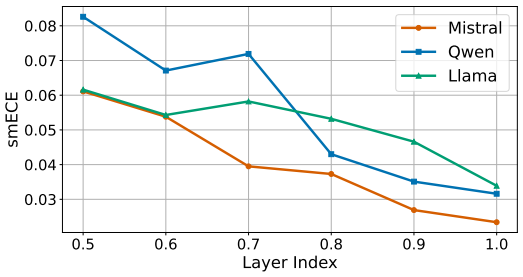

Figure 5: Changes in smECE of PRICE according to the layer index. Representations are extracted from the 50% point of all layers to the final layer for evaluation. Across all three models, smECE tends to decrease toward later layers.

toward later layers, suggesting that confidence-related information becomes more clearly formed in deeper layers. In particular, all models achieve the lowest smECE at the final layer. These results support our design choice of using the final layer for computing PRICE. Results on the full benchmark suite are provided in Table 5 in Appendix C.

7.3 Sensitivity to α

In this section, we compare the theoretically optimal value α^* computed by Eq. (6) with the empirically obtained α. First, we vary the weight α used to compute PRICE from 0 to 1 with a step size of 0.05 and compute smECE for the three models. As shown in Figure 6, the α that empirically minimizes smECE for each model is very close to the theoretical optimum indicated by the dotted line. This indicates that Eq. (6) can effectively estimate the optimal weight while reflecting model-specific characteristics. Overall, the proposed method can

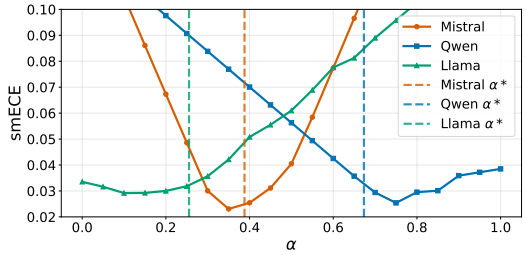

Figure 6: Performance analysis according to the weight α value of PRICE. smECE were measured by varying α from 0 to 1 at 0.05 intervals. The dotted line represents the theoretical optimal α value calculated from Eq. (6).

effectively estimate the optimal hyperparameter without requiring costly search procedures such as grid search.

8 Conclusion

In this study, we proposed PRICE, a framework for estimating query level confidence before LLMs generate answers. PRICE integrates two complementary signals: the structural consensus of attention patterns and the semantic clarity of hidden states. Through extensive experiments across diverse benchmarks and models, PRICE demonstrated superior performance by achieving lower smECE and Brier scroes than existing methods. Furthermore, PRICE mitigates the computational overhead of post-hoc approaches and the generalization limitations of supervised learning-based methods, while enabling both efficiency and safety in LLM systems. These results suggest that the internal representations of LLMs contain sufficiently informative signals for preemptive confidence estimation. Overall, this study demonstrates that effective uncertainty estimation is possible using only internal LLM signals, and highlights their potential for building safer and more efficient LLM systems.

Limitations

This study has several limitations. First, PRICE requires direct access to the model's weights and internal representations. While it can be applied to open source models such as Llama and Mistral, it cannot be applied to closed source models provided in API form. Second, the purpose of PRICE is to estimate confidence, not to guarantee correctness. If the model has bias or receives adversarial input, it can lead to an incorrect answer even if the internal signal appears favorable. Third, while PRICE can warn of low confidence by capturing

the phenomenon of unstable internal signals in the model, it is difficult to specifically identify the reason for low confidence. For example, it is difficult to distinguish whether low confidence is due to the absence of training data or the ambiguity of the input prompt. Fourth, to measure the structural consistency of the attention mechanism, we analyze attention matrices of size $T \times T$. As the length of the input sequence increases, the cost of attention computation and analysis increases significantly. There is a risk of increased memory usage and computational cost in long context situations.

Ethical Statement

This study does not include human participants, personally identifiable information, or sensitive private data. We rely on publicly available research datasets and external resources, such as datasets, pretrained models, and code, while complying with their applicable licenses and terms of use. We acknowledge the original creators of these resources and do not redistribute any third-party materials except where explicitly permitted by their licenses.

References

Elham Asgari, Nina Montaña-Brown, Magda Dubois, Saleh Khalil, Jasmine Balloch, Joshua Au Yeung, and Dominic Pimenta. 2025. A framework to assess clinical safety and hallucination rates of llms for medical text summarisation. *NPJ digital medicine*, 8(1):274.

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.

Jakub Binkowski, Denis Janiak, Albert Sawczyn, Bogdan Gabrys, and Tomasz Jan Kajdanowicz. 2025. Hallucination detection in LLMs using spectral features of attention maps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24354–24385, Suzhou, China. Association for Computational Linguistics.

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

Iván Vicente Moreno Cencerrado, Arnau Padrés Masdemont, Anton Gonzalvez Hawthorne, David Demitri Africa, and Lorenzo Pacchiardi. 2025. No answer needed: Predicting llm answer accuracy from question-only linear probes. *Preprint*, arXiv:2509.10625.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

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

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, Bangkok, Thailand. Association for Computational Linguistics.

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, Mexico City, Mexico. Association for Computational Linguistics.

W Brier Glenn and 1 others. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Hila Gonen, Srini Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148, Singapore. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal,

Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Tobias Groot and Matias Valdenegro Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 145–171, Mexico City, Mexico. Association for Computational Linguistics.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Juyeon Heo, Miao Xiong, Christina Heinze-Deml, and Jaya Narain. 2025. Do LLMs estimate uncertainty well in instruction-following? In *The Thirteenth International Conference on Learning Representations*.

Niall Hurley and Scott Rickard. 2009. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741.

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, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024a. Llm internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 88–104.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024b. LLM internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 88–104, Miami, Florida, US. 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. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

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.

Jianhua Lin. 2002. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Xin Liu, Farima Fatahi Bayat, and Lu Wang. 2024. Enhancing language model factuality via activation-based confidence calibration and guided decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10436–10448, Miami, Florida, USA. Association for Computational Linguistics.

Huan Ma, Jingdong Chen, Guangyu Wang, and Changqing Zhang. 2025. Estimating llm uncertainty with logits. *arXiv e-prints*, pages arXiv–2502.

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

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023b. 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.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.

Frank Nielsen. 2019. On the jensen–shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5):485.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Jahan C Penny-Dimri, Magdalena Bachmann, William R Cooke, Sam Mathewlynn, Samuel Dockree, John Tolladay, Jannik Kossen, Lin Li, Yarin Gal, and Gabriel Davis Jones. 2025. Measuring large language model uncertainty in women's health using semantic entropy and perplexity: a comparative study. *The Lancet Obstetrics, Gynaecology, & Women's Health*, 1(1):e47–e56.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof qa benchmark. *Preprint*, arXiv:2311.12022.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. 2025. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. 2024. On early detection of hallucinations in factual question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2721–2732.

Gaurang Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. 2024. LLM-check: Investigating detection of hallucinations in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for llms. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning models know when they're right: Probing hidden states for self-verification. In *Second Conference on Language Modeling*.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024. Truthx: Alleviating hallucinations by editing large language models in truthful space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8908–8949.

A Baseline Details

Prompt We input the following prompt to the LLM to make it output its own confidence.

Prompt

```
Given the following question, estimate the probability that you would answer it correctly. Do not answer the question. Return only a single real number between 0 and 1, where 0 means completely uncertain and 1 means completely confident.
Question: {question}
Confidence:
```

The temperature of the LLM is set to 0.7. We repeat this procedure three times and record the average value.

Perplexity Perplexity is a method for measuring confidence based on how familiar and predictable an input text is to the LLM. It is computed from the conditional likelihood that the model assigns to the input question text. Let x denote the input question text and T the number of tokens. Then perplexity is defined as follows:

$$\mathrm{PPL}(x) = \exp\left\{-\frac{1}{T}\sum_{t=1}^{T}\log p(x_t|x_{<t})\right\}$$

The underlying assumptions of the perplexity baseline are as follows: (i) Low perplexity: the model finds the input text familiar, and therefore its confidence on the corresponding question is expected to be high. (ii) High perplexity: the model finds the input text unfamiliar, and therefore its confidence on the corresponding question is expected to be low. However, perplexity is generally closer to a measure of the linguistic naturalness of the input sentence or its similarity to the model's pretraining distribution. Thus, it is not a direct indicator of whether the model can actually solve the problem. Nevertheless, prior work has shown that perplexity correlates with downstream task performance (Gonen et al., 2023). Motivated by this observation, we evaluate a perplexity that estimates confidence from the conditional likelihood of the input problem text.

Hidden Score (Sriramanan et al., 2024) Let $\mathbf{H} \in \mathbb{R}^{d\times T}$ denote the hidden representation of the input prompt x at the last layer of the LLM, where d is the hidden dimension and T is the number of tokens. To capture the interactions between token-level hidden representations, we compute the following matrix:

$$\Sigma^2 = \mathbf{H}^\top\mathbf{H}$$

We then compute the log-determinant of this matrix:

$$\log\det(\Sigma^2) = \log\prod_{i=1}^{T}\sigma_i^2 = 2\sum_{i=1}^{m}\log\sigma_i$$

where σ_i denotes the singular values of $\mathbf{H}$. To reduce the dependence on the input prompt length T, we take the average and define the resulting value as the `Hidden Score`:

$$\mathrm{HiddenScore} = \frac{2}{T}\sum_{i=1}^{T}\log\sigma_i$$

Attention Score (Sriramanan et al., 2024) Let a denote the number of attention heads in the last layer of the LLM and T the number of tokens. Then the attention map is a tensor of shape $a \times T \times T$. Let $\mathrm{Ker}_i \in \mathbb{R}^{T\times T}$ denote the attention kernel of each attention head i. In standard transformer self-attention, this kernel is computed as follows:

$$\mathrm{Ker}(Q,K) = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)$$

Due to the causal mask in LLMs, the current token cannot attend to future tokens, and therefore the attention kernel takes the form of a lower triangular matrix. Such a matrix has the property that its eigenvalues are identical to its diagonal elements. Consequently, the log-determinant of Ker_i can be computed directly without resorting to singular value decomposition or eigen decomposition:

$$\log\det(\mathrm{Ker}_i) = \sum_{j=1}^{T}\log\mathrm{Ker}_i^{jj}$$

where Ker_i^{jj} represents the (j,j) diagonal element of the square matrix Ker_i. We compute the mean log-determinant for each head, and define the value aggregated across heads as the `Attention Score`.

Probe1 (Ji et al., 2024b) We extract the hidden state h_T of the last token x_T from the input prompt $\mathbf{x} = (x_1, \cdots, x_T)$ at the last layer of the LLM. This is used to train a multi-layer perceptron (MLP) based classifier, for which we adopt the Llama

Dataset	Count	Source
MMLU (Hendrycks et al., 2021)	14042	https://huggingface.co/datasets/cais/mmlu
BBH (Suzgun et al., 2022)	6511	https://huggingface.co/datasets/maveriq/bigbenchhard
GPQA (Rein et al., 2023)	197	https://huggingface.co/datasets/Idavidrein/gpqa
ARC-C (Clark et al., 2018)	1172	https://huggingface.co/datasets/allenai/ai2:arc
GSM8K (Cobbe et al., 2021)	1319	https://huggingface.co/datasets/openai/gsm8k

Table 2: Dataset information for the five benchmarks used in this study. Count denotes the number of questions used in the experiments, and Source indicates the repository link from which each dataset can be downloaded.

Model	Hidden Dimension	Number of Layers	Source
Mistral-7B-Instruct-v0.3	4096	32	https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
Qwen2.5-7B-Instruct	3584	28	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
Llama-3.1-8B-Instruct	4096	32	https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Table 3: Model information for the three LLMs evaluated in this study. Hidden Dimension denotes the dimensionality of the hidden representations, Number of Layers denotes the number of models layers, and Source indicates the repository link from which each model can be accessed.

Model	MMLU	BBH	GPQA	ARC-C	GSM8K
Mistral7B	57.24	41.24	30.81	78.67	48.37
Qwen7B	71.10	52.13	31.82	90.78	84.00
Llama8B	64.93	45.77	31.82	82.51	55.19

Table 4: Problem-solving accuracy (%) of each LLM on the five benchmark datasets. Accuracy is evaluated using a single greedy-decoding answer for each question.

MLP architecture. The classifier takes h_T as input and predicts whether the answer is correct. We use the probability value output by this classifier as the confidence, that is,

$$\text{Confidence} = P(y = 1 \mid h_T)$$

Probe2 (Liu et al., 2024) From the final layer of the LLM, we extract the hidden state h_i of each token x_i from the input prompt $\mathbf{x} = (x_1, \cdots, x_T)$, and then apply mean pooling to obtain a single vector.

$$z = \frac{1}{T}\sum_{i=1}^{T} h_i$$

We train a classifier that takes z as input to a linear probe and predicts whether the answer is correct or incorrect. The probability output by this classifier is used as the confidence score. That is,

$$\text{Confidence} = \text{sigmoid}(w^\top z + b)$$

B Dataset and Model Information

In this section, we provide detailed information about the benchmark datasets and language models used in this paper. Table 2 reports the number of questions and sources for each benchmark used in our experiments. The benchmarks cover diverse task types and difficulty levels, including commonsense reasoning, science question answering, and mathematical reasoning. This allows us to evaluate confidence estimation performance under different data scales and difficulty conditions.

Table 3 summarizes the model information for the three LLMs evaluated in our experiments. We report the hidden dimension and the number of models layers for each model, since PRICE estimates confidence using internal representations extracted from hidden states and attention patterns. The table also provides the source repository for each model to ensure reproducibility. All models are evaluated under the same experimental setting: for each question, we compute correctness based on the answer obtained from a single greedy decoding run. All benchmarks are evaluated under a 5-shot setting, and the GSM8K benchmark uses Chain-of-Thought prompting.

We conducted our experiments using `transformers` version `5.2.0` and `torch` version `2.10.0`.

C Additional Ablation and Sensitivity Analysis

C.1 Effect of Focus Token Span Length

We analyzed the impact of focus token span length on performance when calculating $S_{\texttt{sal}}$. Experi-

Model	Layer	MMLU		BBH		GPQA		ARC-C		GSM8K	
		smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓
Mistral7B	0.5 (16)	0.0540	0.2478	0.0611	0.2365	0.1386	0.2594	0.1638	0.2522	0.0449	0.2428
	0.6 (20)	0.0535	0.2492	0.0538	0.2341	0.1227	0.2639	0.1528	0.2404	0.0354	0.2158
	0.7 (23)	0.0502	0.2446	0.0395	0.2281	0.1459	0.2536	0.1317	0.2392	0.0388	0.2201
	0.8 (26)	0.0372	0.2448	0.0873	0.2295	0.1493	0.2660	0.1546	0.2271	0.0315	0.2139
	0.9 (29)	**0.0366**	0.2461	0.0269	0.2470	0.1501	0.2625	0.1212	**0.1994**	0.0310	0.2121
	last (32)	0.0371	**0.2399**	**0.0234**	**0.2238**	**0.1139**	**0.2534**	**0.1195**	0.2019	**0.0261**	**0.2020**
Qwen7B	0.5 (14)	0.0899	0.2218	0.0430	0.2460	0.1280	0.2743	0.1687	0.2152	0.1331	0.2253
	0.6 (17)	0.0823	0.2209	0.0671	0.2343	0.1680	0.2847	0.1621	0.1925	0.1111	0.2076
	0.7 (20)	0.0725	0.2075	0.0826	0.2388	0.2208	0.3079	0.1401	0.1893	0.1053	0.2011
	0.8 (23)	0.0702	0.2073	0.0719	0.2365	0.1476	0.2776	0.1296	**0.1847**	0.0879	0.1905
	0.9 (26)	0.0789	0.2147	0.0351	0.2367	0.1107	0.2523	0.1185	0.1937	0.0795	0.1899
	last (28)	**0.0531**	**0.2053**	**0.0316**	**0.2326**	**0.0946**	**0.2316**	**0.0991**	0.1917	**0.0755**	**0.1888**
Llama8B	0.5 (16)	0.0745	0.2249	0.0616	0.2354	0.1702	0.2559	0.1225	0.2260	0.1259	0.2959
	0.6 (20)	0.0628	0.2308	0.0543	0.2349	0.1672	0.2726	0.1189	0.2218	0.1152	0.2899
	0.7 (23)	0.0425	0.2247	0.0582	0.2348	0.1590	0.2579	0.1025	0.1908	0.1089	0.2890
	0.8 (26)	0.0332	0.2260	0.0532	0.2352	0.1515	0.2568	0.0888	0.1934	0.0695	0.2631
	0.9 (29)	0.0260	0.2202	0.0466	0.2453	0.1413	0.2501	0.0759	0.1766	0.0545	0.2582
	last (32)	**0.0257**	**0.2184**	**0.0339**	**0.2341**	**0.1272**	**0.2473**	**0.0428**	**0.1476**	**0.0538**	**0.2510**

Table 5: Layer-wise ablation results of PRICE. We extract attention and hidden states from layers ranging from the 50% depth point to the final layer, and compare smECE and Brier scores. In the Layer column, the value before the parentheses denotes the relative layer position, while the number in parentheses indicates the actual layer index of the model. The best result for each model-dataset pair is marked in **bold**.

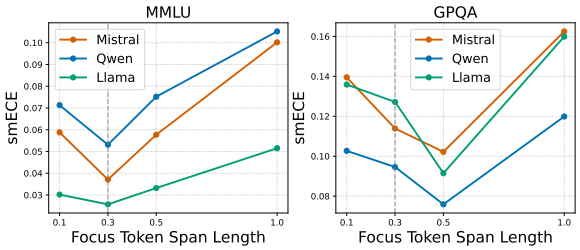

Figure 7: Performance analysis according to focus token span length. The x-axis represents the ratio of how much of the latter part of the input sequence to view. For MMLU, 0.3 shows the lowest error, but for high difficulty tasks like GPQA, a wider region (0.5) tends to be advantageous.

ments were conducted by varying the ratio of latter part tokens relative to the entire sequence to 0.1, 0.3 (proposed), 0.5, and 1.0, with results shown in Figure 7. For benchmarks like MMLU (left chart), the lowest smECE was recorded when set to 0.3, the default setting of PRICE. A narrow region (0.1) is vulnerable to local noise, and the entire region (1.0) dilutes the confidence signal as unnecessary contextual information is mixed in. On the other hand, for the GPQA dataset (right chart), which requires graduate level high difficulty reasoning, a tendency emerged where viewing a slightly wider range (0.5) showed better performance. This can be interpreted as the model going through a more complex reasoning process to reach conclusions for higher difficulty problems, with clues distributed over a wider range.

C.2 Layer-wise Ablation Results

Table 5 reports additional layer-wise smECE and Brier score results across all benchmark datasets. We compute PRICE using attention and hidden states extracted from different layers, ranging from the middle layers to the final layer of each model. In this experiment, we use α^*, where the weight α is computed using Eq. (6).

C.3 Split ratio ablation

When computing PRICE in Figure 3, we use the weight α^* computed by Eq. (6) using 80% of the dataset. In practical deployment scenarios, it is more efficient if the amount of data required to compute α^* is small. We empirically confirm that α^* can achieve sufficient performance even with substantially less data than the 80% ratio used in the main experiments, and that using as little as 20% of the data is sufficient. We vary the proportion of training data as 10%, 20%, 50%, and 80%, where 80% corresponds to the setting used in the main experiments, and compare the resulting

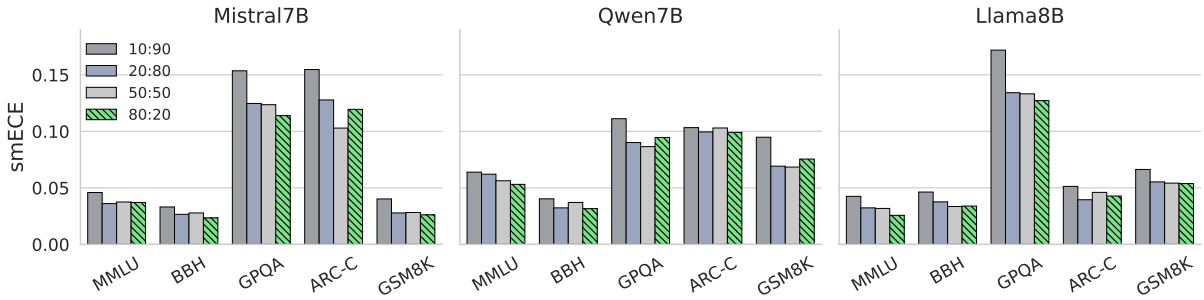

Figure 8: Changes in smECE according to the proportion of training data used to compute α^* using Eq. (6). In the ratio $m : n$, m denotes the proportion of the training dataset. Overall, using only 20% of the training data yields smECE values comparable to those obtained using 50% or 80% of the training data.

smECE values. The results are shown in Figure 8. When only 10% of the data is used, performance becomes somewhat unstable. This instability is particularly pronounced for GPQA, which has a relatively small number of samples. However, when 20% of the data is used, smECE is comparable to the results obtained using 50% or 80% of the data in most cases. We also observe that increasing the proportion of training data does not always lead to improved smECE. These results indicate that α^* computed by Eq. (6) can be determined sufficiently well from a relatively small split.

D Theoretical Details

Proposition 1. JSD_t *defined in Eq. 1 satisfies the following.*

$$0 \leq \mathrm{JSD}_t \leq \log H$$

Equality holds in the following cases.

- $\mathrm{JSD}_t = 0$ *when* $\mathbf{a}_{1,t} = \cdots = \mathbf{a}_{H,t}$
- $\mathrm{JSD}_t = \log H$ *when each head attends to completely different tokens*

Proof. First, we prove $0 \leq \mathrm{JSD}_t$. JSD_t can be expressed in the form of KL divergence (Nielsen, 2019).

$$\mathrm{JSD}_t = \frac{1}{H}\sum_{h=1}^{H} \mathrm{KL}(\mathbf{a}_{h,t} \| \mathbf{m}_t)$$

$$\text{where } \mathrm{KL}(\mathbf{a}_{h,t} \| \mathbf{m}_t) = \sum_k \mathbf{a}_{h,t} \log \frac{\mathbf{a}_{h,t}}{\mathbf{m}_t}$$

Here, k is the index of the key token being attended to. Since KL divergence is always greater than 0 by definition, $\forall h,\ \mathrm{KL}(\mathbf{a}_{h,t} \| \mathbf{m}_t) \geq 0$. Therefore, $\mathrm{JSD}_t \geq 0$. In particular, if $\mathbf{a}_{1,t} = \cdots = \mathbf{a}_{H,t}$, then $\mathbf{m}_t = \mathbf{a}_{1,t}$, so $\forall h,\ \mathrm{KL}(\mathbf{a}_{h,t} \| \mathbf{m}_t) = 0$, and $\mathrm{JSD}_t = 0$.

Next, we prove $\mathrm{JSD}_t \leq \log H$. For each k, the following holds.

$$\mathbf{m}_t(k) = \frac{1}{H}\sum_{h=1}^{H} \mathbf{a}_{h,t}(k) \geq \frac{1}{H}\mathbf{a}_{h,t}(k)$$

Rearranging the above inequality for terms where $\mathbf{a}_{h,t}(k) > 0$ gives the following.

$$\frac{\mathbf{a}_{h,t}(k)}{\mathbf{m}_t(k)} \leq H \ \rightarrow\ \log \frac{\mathbf{a}_{h,t}(k)}{\mathbf{m}_t(k)} \leq \log H$$

Therefore, for each head h, the following holds.

$$\begin{aligned}
\mathrm{KL}(\mathbf{a}_{h,t} \| \mathbf{m}_t) &= \sum_k \mathbf{a}_{h,t} \log \frac{\mathbf{a}_{h,t}}{\mathbf{m}_t} \\
&\leq \sum_k \mathbf{a}_{h,t} \log H \\
&= \log H \sum_k \mathbf{a}_{h,t} \\
&= \log H
\end{aligned}$$

Therefore, the following holds.

$$\begin{aligned}
\mathrm{JSD}_t &= \frac{1}{H}\sum_{h=1}^{H} \mathrm{KL}(\mathbf{a}_{h,t} \| \mathbf{m}_t) \\
&\leq \frac{1}{H}\sum_{h=1}^{H} \log H \\
&= \log H
\end{aligned}$$

Equality holds when $\mathbf{m}_t(k) = \mathbf{a}_{h,t}(k)/H$ for each h, k. This means that if some head h attends to some token position k, the remaining heads assign probability 0 to that position k. □

Proposition 2. *At query position t, define random variables (Z, K) as follows: $Z \sim \mathrm{Unif}\{1, \cdots, H\}$, $K \mid (Z = h) \sim \mathbf{a}_{h,t}$. Then the following holds.*

$$\mathrm{JSD}_t = I(Z; K)$$

Proof. Before the proof, let us redefine several concepts.

Shannon entropy When the distribution of a discrete random variable X is $p(x)$, entropy is defined as follows.

$$\mathcal{H}(X) = -\sum_x p(x) \log p(x)$$

Conditional entropy For two random variables X, Y, conditional entropy is defined as follows.

$$\mathcal{H}(Y \mid X) = \sum_x p(x)\mathcal{H}(Y \mid X = x)$$

Mutual information Mutual information $I(X;Y)$ quantifies how much the uncertainty about Y decreases when X is known, and is defined as follows.

$$I(X;Y) = \mathcal{H}(Y) - \mathcal{H}(Y \mid X)$$

That is, if $I(X;Y) = 0$, then X and Y are independent, and the larger the value, the more strongly they depend on each other.

Consider the random variables (Z, K) in this lemma. By definition, Z is the head index indicating which head was selected, and K is the token index indicating which key position that head selected.

First, for an arbitrary key position $k \in \{1, \cdots, T\}$, applying the law of total probability gives the following.

$$\begin{aligned} P(K = k) &= \sum_{h=1}^{H} P(Z = h)P(K = k \mid Z = h) \\ &= \sum_{h=1}^{H} \frac{1}{H} a_{h,t}(k) \end{aligned}$$

Therefore, the distribution of K is identical to

$$\mathbf{m}_t(k) = \frac{1}{H} \sum_{h=1}^{H} a_{h,k}(k).$$

That is, the marginal distribution of K is $\mathbf{m}_t$, and the following holds.

$$\mathcal{H}(K) = \mathcal{H}(\mathbf{m}_t) \tag{7}$$

Next, we compute the conditional entropy $\mathcal{H}(K \mid Z)$. Since Z follows a uniform distribution, by the definition of conditional entropy, the following holds.

$$\begin{aligned} \mathcal{H}(K \mid Z) &= \sum_{h=1}^{H} P(Z = h)\mathcal{H}(K \mid Z = h) \\ &= \frac{1}{H} \sum_{h=1}^{H} \mathcal{H}(K \mid Z = h) \end{aligned}$$

Here, since $K \mid (Z = h) \sim \mathbf{a}_{h,t}$, we have $\mathcal{H}(K \mid Z = h) = \mathcal{H}(\mathbf{a}_{h,t})$. Therefore, the following holds.

$$\mathcal{H}(K \mid Z) = \frac{1}{H} \sum_{h=1}^{H} \mathcal{H}(\mathbf{a}_{h,t}) \tag{8}$$

Substituting Eq. (7) and Eq. (8) into the definition of mutual information gives the following.

$$\begin{aligned} I(Z;K) &= \mathcal{H}(K) - \mathcal{H}(K \mid Z) \\ &= \mathcal{H}(\mathbf{m}_t) - \frac{1}{H} \sum_{h=1}^{H} \mathcal{H}(\mathbf{a}_{h,t}) \end{aligned}$$

This expression exactly matches the definition of JSD_t defined in Eq. (1). □

In other words, JSD_t represents the amount of information that indicates "how well we can distinguish which head (Z) generated a given key token K when observing it once." If the heads form identical distributions, Z and K become nearly independent, resulting in $I(Z;K) \approx 0$ (high consensus), whereas if each head focuses on different sets of tokens, K carries significant information about Z, leading to an increase in $I(Z;K)$ (large disagreement).

Corollary 1. *The following holds.*

$$\mathcal{H}(K \mid Q) = \mathcal{H}(K \mid H, Q) + I(H; K \mid Q)$$

Therefore, the Consensus Signal can be written as follows.

$$S_{\text{cons}} = 1 - \frac{1}{\log H}\{\mathcal{H}(T|Q) - \mathcal{H}(T|H, Q)\}$$

In the above corollary, each term can be interpreted as follows. $\mathcal{H}(K \mid Q)$ represents the total uncertainty about which key token the model attends to when given query position Q, averaged without distinguishing attention heads. On the other hand, $\mathcal{H}(K \mid H, Q)$ is the conditional uncertainty when the head index is also observed, corresponding to the average entropy of individual attention

heads. Therefore, the difference between these two, $I(H; K \mid Q)$, is a quantity that represents how much the uncertainty about key tokens decreases by additionally observing the attention head information. In other words, the $S_{\texttt{con}}$ can be interpreted as a normalized measure of how effectively the internal consensus of the attention structure reduces uncertainty.

Proposition 3. *Consider $S_{\texttt{con}}$ and $S_{\texttt{sal}}$ as unbiased estimators. That is, $S_{\texttt{con}} = p + \epsilon_1$, $S_{\texttt{sal}} = p+\epsilon_2$, $\mathbb{E}[\epsilon_k|p] = 0$, where p is the (theoretical) true confidence of the model. The variance of each error is $\sigma_1^2 = \mathbb{E}[\epsilon_1^2]$, $\sigma_2^2 = \mathbb{E}[\epsilon_2^2]$, and the covariance is $C = \mathbb{E}[\epsilon_1\epsilon_2]$. Then, the value α^* that minimizes the Brier Score $\mathrm{Brier}(\alpha) = \mathbb{E}[(S_\alpha - Y)^2]$ for $S_{\texttt{PRICE}}$ is as follows.*

$$\alpha^* = \arg\min_\alpha \mathrm{Brier}(\alpha) = \frac{\sigma_2^2 - C}{\sigma_1^2 + \sigma_2^2 - 2C}$$

Proof. For convenience, let $S_{\texttt{con}} = S_1$, $S_{\texttt{sal}} = S_2$ and $S_{\texttt{PRICE}} = S_\alpha$.

$$\begin{aligned} S_\alpha &= \alpha \cdot S_1 + (1-\alpha) \cdot S_2 \\ &= \alpha(p+\epsilon_1) + (1-\alpha)(p+\epsilon_2) \\ &= p + \epsilon_2 + \alpha(\epsilon_1 - \epsilon_2) \end{aligned}$$

Therefore, we can write as follows.

$$S_\alpha - p = \epsilon_2 + \alpha(\epsilon_1 - \epsilon_2) \tag{9}$$

Next, we decompose the Brier score into bias/variance terms.

$$\begin{aligned} \mathrm{BS}(\alpha) &= \mathbb{E}\big[(S_\alpha - Y)^2\big] \\ &= \mathbb{E}\big[\{(S_\alpha - p) + (p - Y)\}^2\big] \\ &= \mathbb{E}\big[(S_\alpha - p)^2\big] + \mathbb{E}\big[(p-Y)^2\big] \\ &\qquad + 2\mathbb{E}\big[(S_\alpha - p)(p - Y)\big] \end{aligned}$$

Here, since $Y|p$ follows $\mathrm{Bernoulli}(p)$, the following holds.

$$Y = p + \eta,\ \mathbb{E}[\eta|p] = 0,\ \mathbb{E}[\eta^2|p] = p(1-p)$$

Therefore, $\mathbb{E}\big[(S_\alpha - p)(p - Y)\big] = \mathbb{E}[(S_\alpha - p)\eta]$, and by definition, if ϵ_1, ϵ_2 are independent of or uncorrelated with p, η, it equals 0. Therefore, the Brier score is as follows.

$$\mathrm{BS}(\alpha) = \mathbb{E}[(S_\alpha - p)^2] + \underbrace{\mathbb{E}[(Y-p)^2]}_{\text{constant independent of } \alpha}$$

Here, the term $\mathbb{E}[(Y-p)^2]$ is a constant independent of α. Therefore, minimizing the Brier can be reduced to the problem of minimizing $\mathbb{E}[(S_\alpha - p)^2]$.

Let us simplify Eq. (9).

$$\begin{aligned} \mathbb{E}[(S_\alpha - p)^2] &= \mathbb{E}[(\epsilon_2 + \alpha(\epsilon_1 - \epsilon_2))^2] \\ &= \sigma_2^2 + 2\alpha\mathbb{E}[\epsilon_2(\epsilon_1 - \epsilon_2)] \\ &\qquad + \alpha^2\mathbb{E}[(\epsilon_1 - \epsilon_2)^2] \end{aligned}$$

Here, the following holds.

$$\begin{aligned} \mathbb{E}[\epsilon_2(\epsilon_1 - \epsilon_2)] &= C - \sigma_2^2 \\ \mathbb{E}[(\epsilon_1 - \epsilon_2)^2] &= \sigma_1^2 + \sigma_2^2 - 2C \end{aligned}$$

Therefore, simplifying $\mathbb{E}[(S_\alpha - p)^2]$ gives the following.

$$f(\alpha) = \alpha^2(\sigma_1^2 + \sigma_2^2 - 2C) + 2\alpha(C - \sigma_2^2) + \sigma_2^2$$

This expression is a quadratic equation in α.

$$f'(\alpha) = 2\alpha(\sigma_1^2 + \sigma_2^2 - 2C) + 2(C - \sigma_2^2)$$

Therefore, the value α^* satisfying $f'(\alpha) = 0$ is as follows.

$$\alpha^* = \frac{\sigma_2^2 - C}{\sigma_1^2 + \sigma_2^2 - 2C} \tag{10}$$

Thus, the proposition is proved. □

However, in practice, it is impossible to use the true confidence p. In the actual implementation, we derive the closed-form solution from the training split.

Proposition 4 (Closed-form solution for finite samples)**.** *Let $S_{\texttt{con},i}$, $S_{\texttt{sal},i}$, $y_i \in \{0,1\}$ denote the consensus signal, salience signal, and correctness label for each sample i, respectively. Then the optimal weight α^* satisfies*

$$\alpha^* = \arg\min_\alpha \frac{1}{n}\sum_{i=1}^{n}\big\{\alpha S_{1,i} + (1-\alpha)S_{2,i} - y_i\big\}^2. \tag{11}$$

The corresponding α^* can be written in the following closed form:

$$\alpha^* = \frac{\sum_{i=1}^n (S_{1,i} - S_{2,i})(y_i - S_{2,i})}{\sum_{i=1}^n (S_{1,i} - S_{2,i})^2}. \tag{12}$$

Proof. For notational convenience, we write $S_{\texttt{con},i} = S_{1,i}$ and $S_{\texttt{sal},i} = S_{2,i}$. Then the optimal weight α is given by

$$\alpha^* = \arg\min_\alpha \frac{1}{n}\sum_{i=1}^{n}\big\{\alpha S_{1,i} + (1-\alpha)S_{2,i} - y_i\big\}^2. \tag{13}$$

The term inside the square can be rewritten as $\alpha S_{1,i} + (1-\alpha)S_{2,i} - y_i = \alpha(S_{1,i} - S_{2,i}) - (y_i - S_{2,i})$. For convenience, let $S_{1,i} - S_{2,i} = x_i$ and $y_i - S_{2,i} = r_i$. Then α^* can be written as

$$\alpha^* = \arg\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} (\alpha x_i - r_i)^2. \tag{14}$$

Let $f(\alpha) = \sum_{i=1}^{n} (\alpha x_i - r_i)^2$. Then

$$\begin{aligned} f(\alpha) &= \sum_{i=1}^{n} (\alpha x_i - r_i)^2 \\ &= \sum_{i=1}^{n} (\alpha^2 x_i^2 - 2\alpha x_i r_i + r_i^2) \\ &= \alpha^2 \sum_{i=1}^{n} x_i^2 - 2\alpha \sum_{i=1}^{n} x_i r_i + \sum_{i=1}^{n} r_i^2. \end{aligned}$$

The function f is a quadratic function of α. Therefore, $f(\alpha)$ attains its minimum when $f'(\alpha) = 0$. Differentiating $f(\alpha)$ with respect to α gives

$$f'(\alpha) = 2\alpha \sum_{i=1}^{n} x_i^2 - 2 \sum_{i=1}^{n} x_i r_i.$$

The value of α satisfying $f'(\alpha) = 0$ is

$$\alpha = \frac{\sum_{i=1}^{n} x_i r_i}{\sum_{i=1}^{n} x_i^2}. \tag{15}$$

Therefore, the closed-form solution to Eq. (13) is

$$\alpha^* = \frac{\sum_{i=1}^{n} (S_{1,i} - S_{2,i})(y_i - S_{2,i})}{\sum_{i=1}^{n} (S_{1,i} - S_{2,i})^2}.$$

□

By replacing the empirical averages in this expression with expectations, we obtain

$$\alpha^* = \frac{\mathbb{E}[(S_1 - S_2)(Y - S_2)]}{\mathbb{E}[(S_1 - S_2)^2]}.$$

Applying the assumptions stated in Proposition 3, the population-level solution above can be expressed as Eq. (10).

Lemma 1 (smECE is controlled by the error to true confidence)**.** *Consider a non-negative smoothing kernel $K_\sigma(t, f) \geq 0$, and define the population-level smECE as follows:*

$$\mathrm{smECE}_\sigma(S_\alpha) = \int_0^1 |r_{\sigma,\alpha}(t)| \delta_{\sigma,\alpha}(t) dt. \tag{16}$$

Here, $r_{\sigma,\alpha}(t)$ and $\delta_{\sigma,\alpha}(t)$ are defined as

$$\begin{aligned} r_{\sigma,\alpha}(t) &= \frac{\mathbb{E}[K_\sigma(t, S_\alpha)(Y - S_\alpha)]}{\mathbb{E}[K_\sigma(t, S_\alpha)]} \\ \delta_{\sigma,\alpha}(t) &= \mathbb{E}[K_\sigma(t, S_\alpha)]. \end{aligned}$$

Assume that there exists a constant C_K such that

$$\sup_{f \in [0,1]} \int_0^1 K_\sigma(t, f) dt \leq C_K < \infty. \tag{17}$$

Then, for all $\alpha \in [0, 1]$, the following holds:

$$\mathrm{smECE}_\sigma(S_\alpha) \leq C_K \sqrt{\mathbb{E}[(S_\alpha - p)^2]}. \tag{18}$$

The assumption in Eq. (17) is naturally satisfied by standard smoothing kernels. For example, suppose we use a normalized kernel satisfying

$$\begin{aligned} K_\sigma(t, f) &= \sigma^{-1} K((t - f)/\sigma) \\ \int_{\mathbb{R}} K(u) du &= 1. \end{aligned}$$

Then,

$$\int_0^1 K_\sigma(t, f) dt \leq \int_{-\infty}^{\infty} K_\sigma(t, f) = 1.$$

In this case, we can set $C_K = 1$.

Proof. We start from the definition of smECE. Substituting the definitions of r and δ into Eq. (16), smECE can be written as

$$\int_0^1 \left| \frac{\mathbb{E}[K_\sigma(t, S_\alpha)(Y - S_\alpha)]}{\mathbb{E}[K_\sigma(t, S_\alpha)]} \right| \mathbb{E}[K_\sigma(t, S_\alpha)] dt.$$

If $\mathbb{E}[K_\sigma(t, S_\alpha)] > 0$, then smECE becomes

$$\int_0^1 |\mathbb{E}[K_\sigma(t, S_\alpha)(Y - S_\alpha)]| dt. \tag{19}$$

Since $Y \,|\, p$ follows Bernoulli(p), we can write $Y = p + \eta$. Then $Y - S_\alpha = (p - S_\alpha) + \eta$. Substituting this into the integrand of Eq. (19) gives

$$\begin{aligned} &\mathbb{E}[K_\sigma(t, S_\alpha)(Y - S_\alpha)] = \\ &\quad \mathbb{E}[K_\sigma(t, S_\alpha)(p - S_\alpha)] + \mathbb{E}[K_\sigma(t, S_\alpha)\eta]. \end{aligned}$$

We now show that the second term $\mathbb{E}[K_\sigma(t, S_\alpha)\eta]$ is zero. Since S_α is a function of $S_{\mathtt{con}}$ and $S_{\mathtt{sal}}$, $K_\sigma(t, S_\alpha)$ is also measurable with respect to $p, S_{\mathtt{con}}$, and $S_{\mathtt{sal}}$. Applying the law of total expectation yields

$$\mathbb{E}[K_\sigma(t, S_\alpha)\eta] = \mathbb{E}[K_\sigma(t, S_\alpha)\, \mathbb{E}[\eta \,|\, p, S_{\mathtt{con}}, S_{\mathtt{sal}}]].$$

By assumption, $\mathbb{E}[\eta \mid p, S_{\texttt{con}}, S_{\texttt{sal}}] = 0$. Therefore, the second term satisfies $\mathbb{E}[K_\sigma(t, S_\alpha)\eta] = 0$. Thus, Eq. (19) can be written as

$$\mathrm{smECE}_\sigma(S_\alpha) = \int_0^1 |\mathbb{E}[K_\sigma(t, S_\alpha)(p - S_\alpha)]|dt.$$

Applying the triangle inequality to $|\mathbb{E}[K_\sigma(t, S_\alpha)(p - S_\alpha)]|$, we obtain

$$\mathrm{smECE}_\sigma(S_\alpha) \leq \int_0^1 \mathbb{E}[K_\sigma(t, S_\alpha)|p - S_\alpha|]dt.$$

We then use Tonelli's theorem. In simple terms, Tonelli's theorem implies that

$$\int \mathbb{E}[g(t, X)]dt = \mathbb{E}\left[\int g(t, X)dt\right].$$

Applying this theorem gives

$$\mathrm{smECE}_\sigma(S_\alpha) \leq \mathbb{E}\Big[|p - S_\alpha| \int_0^1 K_\sigma(t, S_\alpha)dt\Big].$$

By the assumption in Eq. (17), we have

$$\mathrm{smECE}_\sigma(S_\alpha) \leq C_K \cdot \mathbb{E}[|p - S_\alpha|].$$

Finally, applying the Cauchy-Schwarz inequality yields

$$\begin{aligned}
&\mathbb{E}[|p - S_\alpha|] \\
&= \mathbb{E}[|p - S_\alpha| \cdot 1] \leq \sqrt{\mathbb{E}[(p - S_\alpha)^2] \cdot \mathbb{E}[1^2]} \\
&= \sqrt{\mathbb{E}[(p - S_\alpha)^2]}.
\end{aligned}$$

This completes the proof of Eq. (18). □

Corollary 2 (Brier optimization controls smECE)**.** *Suppose that the assumptions in Lemma 1 and Proposition 3 hold. Then, the weight α^* that minimizes the Brier score* $\mathrm{Brier}(\alpha) = \mathbb{E}[(S_\alpha - Y)^2]$ *in Eq. (10) also minimizes the upper bound of smECE,* $C_K\sqrt{\mathbb{E}[(S_\alpha - Y)^2]}$.

This corollary does not claim that the weight α^* minimizing the Brier score directly minimizes smECE itself. Rather, it means that α^* minimizes a theoretical upper bound of smECE.

Lemma 2 (Finite-sample smECE is upper bounded by the empirical Brier score)**.** *Define the empirical smECE as*

$$\widehat{\mathrm{smECE}}_\sigma(S_\alpha) = \int_0^1 |\hat{r}_\sigma(t)|\hat{\delta}_\sigma(t)dt.$$

Here, $\hat{r}_\sigma(t)$ and $\hat{\delta}_\sigma(t)$ are defined as follows:

$$\hat{r}_\sigma(t) = \frac{\sum_{i=1}^n K_\sigma(t, s_i)(y_i - s_i)}{\sum_{i=1}^n K_\sigma(t, s_i)}$$

$$\hat{\delta}_\sigma(t) = \frac{1}{n}\sum_{i=1}^n K_\sigma(t, s_i).$$

As in the population case, assume that there exists a constant C_K such that

$$\sup_{s\in[0,1]} \int_0^1 K_\sigma(t, s)dt \leq C_K < \infty.$$

Then, the following holds:

$$\widehat{\mathrm{smECE}}_\sigma(S_\alpha) \leq C_K\sqrt{\widehat{\mathrm{Brier}}(\alpha)}.$$

The proof follows the same argument as Lemma 1.

E Analysis on head importance weighting

The multi head attention mechanism in transformers is designed to enable the model to simultaneously learn information from different positions and various representational subspaces (Vaswani et al., 2017). Accordingly, each head tends to specialize in different domains, such as capturing syntactic structures or semantic relationships (Clark et al., 2019). However, prior research has pointed out that only a small number of heads make decisive contributions to the inference process, while the remaining heads contain redundant or unimportant information and can be removed (Voita et al., 2019; Michel et al., 2019).

This perspective raises the hypothesis that treating all heads equally when estimating confidence may not be optimal. There is a possibility that selectively focusing on key heads that substantially contribute to predictions could improve the accuracy of confidence estimation. To verify this hypothesis, we conducted additional comparative experiments by introducing weights that reflect the actual contribution of each head.

In the transformer architecture, the degree to which a specific head h contributes to the residual stream is proportional not to the attention probability values themselves, but to the magnitude of the output vector computed as a result (Kobayashi et al., 2020). Let $\mathbf{o}_{h,t} \in \mathbb{R}^d$ denote the value output vector of head h. We define the importance $w_{h,t}$ of that head at time step t as the ratio of L2 norms as

Model	Method	MMLU		BBH		GPQA		ARC-C		GSM8K	
		smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓
Mistral7B	Head weight	0.0788	0.2689	0.0683	0.2198	0.1196	0.2187	0.1357	0.1867	0.0759	0.2508
	No weight	**0.0371**	**0.1999**	**0.0234**	**0.1838**	**0.1139**	**0.2134**	**0.1195**	**0.1819**	**0.0261**	**0.2020**
Qwen7B	Head weight	0.0937	0.2233	0.0652	0.2455	0.1232	0.2603	0.1052	0.1723	**0.0698**	**0.1682**
	No weight	**0.0531**	**0.1653**	**0.0316**	**0.1926**	**0.0946**	**0.2216**	**0.0991**	**0.1388**	0.0755	0.1888
Llama8B	Head weight	0.0699	0.2151	0.0498	0.2102	0.1564	0.2511	0.0876	0.1566	0.1033	0.2599
	No weight	**0.0257**	**0.1784**	**0.0339**	**0.1941**	**0.1272**	**0.2073**	**0.0428**	**0.1276**	**0.0538**	**0.2110**

Table 6: The following are the experimental results comparing the conventional method (no weighting) with a method that assigns weights to attention heads based on their importance. In the majority of scenarios, the unweighted approach demonstrated superior performance. Consequently, we adopted a configuration that assigns equal weights to all heads as our primary setting.

follows.

$$w_{h,t} = \frac{\|\mathbf{o}_{h,t}\|_2}{\sum_{j=1}^{H} \|\mathbf{o}_{j,t}\|_2}, \quad \text{where} \sum_{h=1}^{H} w_{h,t} = 1$$

This weighting scheme naturally excludes the influence of unnecessary heads that do not substantially contribute to the model's predictions.

Table 6 shows the impact of weighting based on attention head importance on confidence estimation performance. As in the main experiments, we evaluated four models across five benchmarks, and the results showed that the baseline approach (No weight) achieved lower smECE and Brier scores than the head weight approach, demonstrating overall superior performance. In some cases (e.g., GPQA for Mistral, GSM8K and ARC for Qwen), the head weight setting showed marginally better performance. Particularly for the challenging GPQA benchmark, the Mistral and Llama models performed better with the head weight setting. This suggests that specific heads may play important roles in complex reasoning processes, but in general situations, uniformly considering all heads proves to provide more robust performance.

These results suggest that for confidence estimation, it is important to comprehensively consider all heads, not just a few key heads. While previous studies assigned importance assuming that specific heads dominate information processing, in the case of confidence estimation, signals from heads with relatively low importance also serve as useful cues. In other words, applying weights risks suppressing some subtle signals, which can hinder overall confidence estimation. Therefore, our PRICE framework adopts an approach that computes the average of JSD across all heads without separate weight calculation. This has lower computational complexity than head weighting and guarantees stable performance across various settings.

Model	Method	MMLU		BBH		GPQA		ARC-C		GSM8K	
		smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓	smECE ↓	Brier ↓
Mistral7B	Prompt	0.3689	0.3255	0.3088	0.3357	0.4111	0.3988	0.2989	0.3041	0.3201	0.3556
	Perplexity	0.2655	0.3303	0.5579	0.5520	0.5303	0.5052	**0.0846**	**0.1796**	0.4115	0.5163
	Hidden Score	0.1001	0.2478	0.1028	0.2389	0.3460	0.3547	0.0988	0.1829	0.1315	0.2972
	Attention Score	0.1017	0.2608	0.1829	0.2720	0.4018	0.3959	0.1767	0.2097	0.0764	0.2511
	PRICE ($\alpha = 0.5$)	**0.0592**	**0.2295**	**0.0402**	**0.1989**	**0.1566**	**0.2555**	0.1399	0.1901	**0.0633**	**0.2348**
	PRICE (α^*)	0.0371	0.1999	**0.0234**	**0.1838**	0.1139	**0.2134**	0.1195	0.1819	**0.0261**	**0.2020**
	Probe1 (Ji et al., 2024b)	**0.0327**	**0.1848**	0.0266	0.1997	0.1074	0.2228	**0.0523**	0.1440	0.0318	0.2365
	Probe2 (Liu et al., 2024)	0.0336	0.1896	0.0279	0.2011	**0.1053**	0.2233	0.0698	**0.1389**	0.0395	0.2352
Qwen7B	Prompt	0.1958	0.2296	0.1913	0.2196	0.3628	0.3508	0.2115	0.2443	0.4189	0.4336
	Perplexity	0.1784	0.2434	0.4717	0.4722	0.5315	0.5111	0.1259	**0.1114**	0.4255	0.5092
	Hidden Score	0.0864	0.2165	0.0875	0.2523	0.3122	0.3391	0.2226	0.1510	0.1515	0.2709
	Attention Score	0.0983	0.2126	**0.0542**	0.2452	0.2674	0.3114	0.2293	0.1571	0.1043	**0.2161**
	PRICE ($\alpha = 0.5$)	**0.0531**	**0.1653**	0.0588	**0.2188**	**0.1292**	**0.2696**	**0.1203**	0.1659	**0.1038**	0.2296
	PRICE (α^*)	0.0531	0.1653	**0.0316**	**0.1926**	**0.0946**	**0.2216**	0.0991	0.1917	0.0755	0.1888
	Probe1 (Ji et al., 2024b)	**0.0211**	**0.1493**	0.0504	0.2020	0.1119	0.2426	**0.0277**	**0.0708**	**0.0325**	**0.1274**
	Probe2 (Liu et al., 2024)	0.0298	0.1534	0.0488	0.1976	0.1198	0.2493	0.0539	0.1055	0.0489	0.1331
Llama8B	Prompt	0.1798	0.1969	0.2548	0.3099	0.4518	0.4885	0.1295	0.1573	0.4259	0.3933
	Perplexity	0.2272	0.2914	0.5211	0.5199	0.5620	0.5310	0.1329	0.1665	0.2902	0.3119
	Hidden Score	0.0610	0.2256	0.1658	0.2642	0.3719	0.3746	0.0887	0.1574	0.1077	0.2987
	Attention Score	0.0792	0.2405	0.1743	0.2697	0.2819	0.3248	0.0731	0.1558	0.0823	0.2621
	PRICE ($\alpha = 0.5$)	**0.0603**	**0.1953**	**0.0591**	**0.2139**	**0.1596**	**0.2563**	**0.0709**	**0.1399**	**0.0783**	**0.2452**
	PRICE (α^*)	0.0257	0.1784	0.0339	**0.1941**	**0.1272**	**0.2073**	0.0428	0.1276	**0.0538**	**0.2110**
	Probe1 (Ji et al., 2024b)	**0.0121**	**0.1737**	**0.0231**	0.2079	0.1904	0.2515	**0.0318**	**0.1193**	0.1489	0.2409
	Probe2 (Liu et al., 2024)	0.0416	0.1905	0.0302	0.1991	0.1318	0.2198	0.0395	0.1253	0.0698	0.2121

Table 7: Confidence estimation performance measured across three LLMs and five datasets. Lower smECE and Brier score values indicate better performance. The best performance is marked in **bold**, and the second-best performance is underlined. Gray shading indicates results compared with training-based methods.