ADVICE: Answer-Dependent Verbalized Confidence Estimation

Anonymous ACL submission

Abstract

Recent progress in large language models (LLMs) has enabled them to communicate their confidence in natural language, improving transparency and reliability. However, this expressiveness is often accompanied by systematic overconfidence, whose underlying causes remain poorly understood. In this work, we analyze the dynamics of verbalized confidence estimation and identify *answer-independence*—the failure to condition confidence on the model's own answer—as a primary driver of this behavior. To address this, we introduce **ADVICE** (**A**nswer-**D**ependent **V**erbal**I**zed **C**onfidence **E**stimation), a fine-tuning framework that promotes answer-grounded confidence estimation. Extensive experiments show that ADVICE substantially improves confidence calibration, while exhibiting strong generalization to unseen settings without degrading task performance. We further demonstrate that these gains stem from enhanced answer dependence, shedding light on the origins of overconfidence and enabling trustworthy confidence verbalization.

1 Introduction

Recent advances in large language models (LLMs) have led to improvements in performance across diverse tasks (Grattafiori et al., 2024; OpenAI et al., 2024). Nonetheless, hallucination—the generation of factually inaccurate or fabricated content—remains a persistent limitation (Ji et al., 2023), with some arguing that it is theoretically unavoidable (Xu et al., 2024; Kalai et al., 2025). This poses an obstacle to the reliable use of LLMs, particularly in high-stakes domains such as law and healthcare (Jayakumar et al., 2023; Sakai and Lam, 2025).

As a remedy, recent studies refine LLMs to provide not only answers but also confidence estimates (Lin et al., 2022; Tian et al., 2023; Xiong et al., 2024), aiming to manage the inherent incompleteness of LLMs rather than eliminate it entirely. In this sense, the estimated confidence is intended to approximate the likelihood of the corresponding answer being correct (Guo et al., 2017).[1] Well-calibrated models can thus express high assurance when confident and appropriately convey caution when uncertain, reinforcing their reliability.

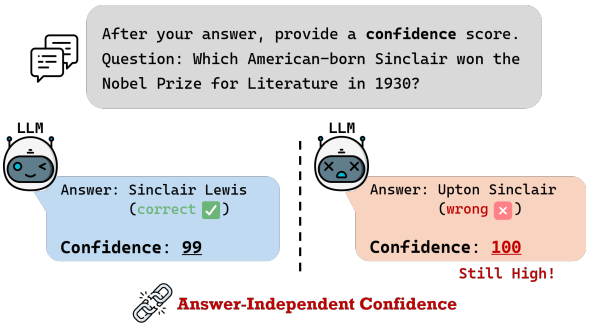


Figure 1: LLMs tend to verbalize their overconfidence *irrespective of whether their answers are correct*. We propose a method (ADVICE) to mitigate this problem, achieving well-calibrated verbalized confidence.

Confidence estimation in LLMs has been explored through a range of approaches, including post-hoc extraction of confidence scores. Among these, *verbalized confidence*, which requires LLMs to articulate confidence levels in natural language during generation, has attracted sustained attention due to its universal applicability and user-friendly nature (Yang et al., 2025). However, its broader application is hindered by the well-known issue of overconfidence (Xiong et al., 2024; Groot and Valdenegro Toro, 2024; Leng et al., 2025; Xu et al., 2025), i.e., the tendency to assign high confidence irrespective of output quality (see Figure 1).

In the literature, research on mitigating the overconfidence problem can be broadly categorized into three directions: prompting-based techniques, sampling-based methods such as self-consistency

[1] In related work, the terms **uncertainty** and **confidence** are often used interchangeably. For clarification, we follow the definitions of Lin et al. (2024): uncertainty pertains only to the input (q), i.e., $P(\cdot|q)$, while confidence concerns both the input and the corresponding answer (a), that is, $P(\cdot|q, a)$.

(Zhou et al., 2025), and fine-tuning (Li et al., 2025). Although such methods have contributed to improved calibration, their emphasis lies on *how to mitigate overconfidence rather than why it arises*, leaving its primary causes largely unexplained.

In this work, we first investigate the intermediate process through which LLMs estimate confidence, eliciting explicit verbalization and probing their inner workings. Specifically, we study how much the model relies on its own answer, since this property characterizes confidence and differentiates it from other measures of uncertainty (Footnote 1).[2] Our analyses reveal that LLM-generated answers and confidence verbalization seem to be internally decoupled, implying that this disjunction may underlie the poor calibration of verbalized confidence.

To further study the role of answer-groundedness in verbalized confidence estimation, we propose a novel method, **ADVICE** (**A**nswer-**D**ependent **V**erbal**I**zed **C**onfidence **E**stimation). ADVICE explicitly encourages the model to focus more on its answer when reporting its confidence, serving as a barometer for evaluating the answer's influence.

Through experiments, we demonstrate that ADVICE achieves performance comparable to state-of-the-art sampling-based and fine-tuning–based methods, confirming the importance of answer information in confidence estimation. Moreover, ADVICE offers several advantages: (1) improved confidence calibration with strong generalization, (2) compact and efficient confidence representation, and (3) no compromise in task performance.

Lastly, we revisit our initial internal analysis with ADVICE-enhanced LLMs and show that performance gains are causally driven by stronger answer dependence, supporting our central claim.

2 Related Work

Verbalized confidence Since Lin et al. (2022) introduced verbalized confidence estimation, numerous studies have explored its potential, highlighting its model-agnostic design, cost-effectiveness, and accessibility to model knowledge (Yang et al., 2025). In particular, a broad spectrum of work has sought to improve its calibration. As an initial direction, post-hoc methods that do not require model modification—such as prompting-based and sampling-based ones (Zhao et al., 2024; Yang et al., 2025; Zhou et al., 2025)—have been proposed. In contrast, several studies (Tian et al., 2023; Stangel et al., 2025; Li et al., 2025) adopt fine-tuning methods, specifically for the task of question answering (QA). However, prior studies have mainly centered on developing new methods for achieving quantitative improvements, with limited qualitative analysis of the underlying mechanisms. To fill this gap, we present an in-depth investigation into the operational mechanism of verbalized confidence estimation and introduce an intuitive method.

LLM probing methods With the wide adoption of LLMs, understanding their inner workings has become crucial, leading to a surge of research on their mechanistic interpretability (Mohammadi et al., 2025). In particular, a line of work on controlling and analyzing the attention mechanism, e.g., Attention Rollout, Attention Flow, and Attention Knockout (Abnar and Zuidema, 2020; Geva et al., 2023), has gained interest. Meanwhile, gradient-based attribution methods provide a more direct quantification of output sensitivity to input perturbations. Integrated Gradients (Sundararajan et al., 2017) attributes output importance to input tokens by integrating gradients along the path from a baseline to the input. We employ methods from both paradigms to probe the relationship between verbalized confidence estimation and the model's answer.

3 Claim: Verbalized Confidence is Nearly Answer-Independent

By definition, verbalized confidence should reflect a model's degree of belief in its generated answer. To examine whether this causal relationship holds in practice, we perform two evaluations: (1) a comparison of confidence distributions conditioned on alternative answer candidates, and (2) an attribution-based analysis. The results reveal that surprisingly and counterintuitively, verbalized confidence is nearly independent of the answer.

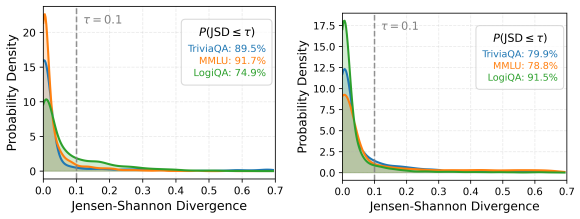
3.1 Comparison of Confidence Distributions

Let $q \in Q$ represent a question, and A_q indicates the set of all possible answer predictions for the given question, including both factually correct and incorrect ones. C denotes the set of confidence expressions, such as `0` (very low) to `9` (very high).[3]

We investigate whether verbalized confidence is

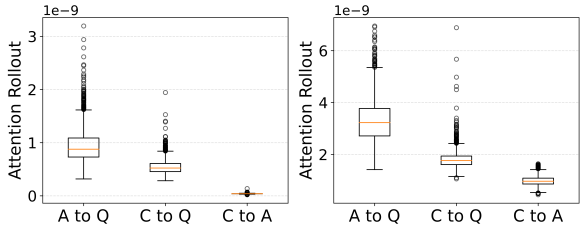
[2] We further take inspiration from neuroscience (Navajas et al., 2016; Desender et al., 2021), where confidence estimation is framed as post-decisional evidence accumulation.

[3] We assume that the model expresses confidence in a discrete form using numerical or verbal tokens (refer to §5).



(a) GEMMA2-9B-IT (b) LLAMA3.1-8B-INSTRUCT

Figure 2: Probability density functions (PDFs) of the set $\{\text{JSD}(P_M(C|q,a_i)||P_M(C|q,a_j))\}$, where $M \in \{\text{GEMMA2}, \text{LLAMA3.1}\}$ and (q, a_i, a_j) are from TriviaQA, MMLU, LogiQA. Each PDF (solid curve) is computed via Gaussian kernel density estimation. Near-zero concentration implies answer-independent confidence.



(a) GEMMA2-9B-IT (b) LLAMA3.1-8B-INSTRUCT

Figure 3: Comparison of Attention Rollout scores on three attention directions: (1) Answer to Question, (2) Confidence to Question, and (3) Confidence to Answer.

independent of the answer by testing the following:

$$P_M(C \mid q, a_i) \approx P_M(C \mid q, a_j) \;\; \forall a_i \neq a_j \in A_q,$$

where $P_M(C \mid \cdot)$ represents the probability distribution over confidence expressions computed by the model M.[4] If the above equation holds, the Jensen–Shannon divergence (JSD) (Menéndez et al., 1997) between the left-hand side (LHS) and the right-hand side (RHS) should approach zero:

$$\text{JSD}(P_M(C \mid q, a_i) \;||\; P_M(C \mid q, a_j)) \approx 0.$$

To characterize trends in JSD values across combinations of q, a_i, and a_j, we compute $\sum_{q \in Q} \binom{|A_q|}{2}$ JSD scores for three datasets (TriviaQA, MMLU, LogiQA) and visualize their distributions. We apply this process to GEMMA-2-9B-IT and LLAMA-3.1-8B-INSTRUCT (see Appendix A for details).

In Figure 2, we observe (1) strong concentrations of JSD scores near zero with long right tails and (2) high densities of samples in the region where JSD ≤ 0.1.[5] Overall, these results suggest that confidence estimates vary minimally across different answers, indicating limited use of answer-specific information. Results under extra experimental settings are reported in Figure 10, 11, and 12.

3.2 Attribution-Based Analysis

While the findings in §3.1 are striking, they warrant further validation through additional evidence from alternative analytical perspectives. To this end, we employ two attribution methods: Attention Rollout and Integrated Gradients.

Attention Rollout Attention Rollout (AR) (Abnar and Zuidema, 2020) quantifies the contribution of input tokens to model predictions by recursively aggregating attention weights across layers.[6] We use AR to analyze how different components of the input prompt—the question (Q), answer (A), and verbalized confidence (C)—interact through attention inside the model. Specifically, we examine attention from C to A ($C \rightarrow A$) and compare its average AR score against other attention flows, such as $A \rightarrow Q$ and $C \rightarrow Q$. As shown in Figure 3, the AR score of $C \rightarrow A$ is significantly lower than those of the reference cases, suggesting that LLMs rely less on answer-specific information when estimating confidence.

Integrated Gradients While Attention Rollout captures attention-level interactions, Integrated Gradients (IG) provides a gradient-based perspective that enables a qualitative analysis of how different input components contribute to verbalized confidence. Figure 13 in the Appendix presents the attribution scores assigned to individual input tokens. We observe that answer tokens are consistently under-weighted compared to tokens in other components, such as "`user`" and the BOS token.

Takeaways Our empirical analyses confirm that the generation of verbalized confidence operates substantially independently of cues from the answer component, contrary to its intended definition. As a result, we argue that this phenomenon constitutes a primary factor underlying poor calibration and overconfidence in verbalized confidence.

4 ADVICE: Answer-Dependent Verbalized Confidence Estimation

We present **ADVICE** (see Figure 4), a lightweight training framework designed to reinforce answer-

[4]For computational efficiency, we restrict A_q to 30 answers generated by M using top-p sampling.

[5]We set the threshold $\tau = 0.1$ to compute the fraction of samples with near-zero divergence, following prior work (Milan Kummaya et al., 2025; Deho et al., 2025) that defines two distributions as similar when their JSD is ≤ 0.1.

[6]We refer readers to Appendix B for algorithmic details.

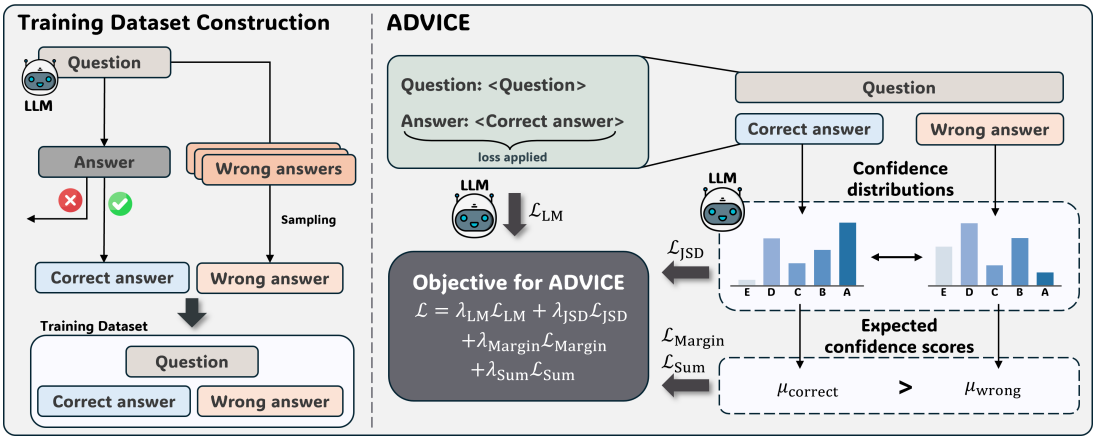


Figure 4: Illustration of the proposed **ADVICE** (**A**nswer- **D**ependent **V**erbal**I**zed **C**onfidence **E**stimation) framework.

groundedness in verbalized confidence estimation.

4.1 Training Dataset Construction

We adopt TriviaQA (Joshi et al., 2017) as our training dataset, which is an open-domain, free-form question answering benchmark. We begin by sampling 4,000 instances from the training split of the dataset. Subsequently, we retain only instances where the model generates the correct answer under greedy decoding. To encourage the model to express high confidence for correct answers and low confidence for incorrect ones, we construct, for each question q, a pair consisting of a correct answer (a_{correct}) and an incorrect answer (a_{wrong}), yielding a triplet $(q, a_{\text{correct}}, a_{\text{wrong}})$. The incorrect answer ($a_{\text{wrong}}$) is randomly sampled from the model's responses using stochastic decoding. Finally, as verbalized confidence can appear in various formats as described in §5, we construct two variants for each instance to train a model capable of fluently expressing confidence in multiple forms. Examples are provided in Table 7 in the Appendix.

4.2 Training Objectives

Motivated by our findings in §3, we train the model to explicitly condition its confidence on the generated answer while preserving its performance on general tasks. Specifically, we define four training objectives for each triplet obtained from §4.1:

$$\mathcal{L}_{\text{LM}} = \frac{1}{|a_{\text{correct}}|} \sum_{x_t \in a_{\text{correct}}} -\log P(x_t \mid x_{<t}),$$

$$\mathcal{L}_{\text{JSD}} = \max(0, \delta_{\text{JSD}} - D_{\text{JSD}}(P_{\text{correct}} \mid\mid P_{\text{wrong}})),$$

$$\mathcal{L}_{\text{Margin}} = \max(0, \delta_{\text{Margin}} - (\mu_{\text{correct}} - \mu_{\text{wrong}})),$$

$$\mathcal{L}_{\text{Sum}} = |1 - (\mu_{\text{correct}} + \mu_{\text{wrong}})|,$$

where $\mathcal{L}_{\text{LM}}$ denotes the negative log-likelihood of the correct answer a_{correct}, added to preserve general task (e.g., QA) abilities as in Li et al. (2025). $\mathcal{L}_{\text{JSD}}$ explicitly drives the model to learn contrasting confidence distributions (P_{correct} and P_{wrong}) for the correct (a_{correct}) and wrong answers (a_{wrong}) given the same question q. However, $\mathcal{L}_{\text{JSD}}$ provides no directional constraint, implying that it may still converge even if the model erroneously assigns greater confidence to incorrect answers while underestimating correct ones. To resolve this, we apply $\mathcal{L}_{\text{Margin}}$, formulated as the difference between the expected confidence assigned to correct answers (μ_{correct}) and that assigned to incorrect ones (μ_{wrong}). Additionally, we introduce $\mathcal{L}_{\text{Sum}}$ to enforce the ideal constraint $\mu_{\text{correct}} + \mu_{\text{wrong}} = 1$, reflecting that confidence should represent the likelihood of the answer being correct (Guo et al., 2017). Hyperparameters δ_{JSD} and δ_{Margin} are set to control the extent to which the model differentiates between correct and incorrect answers.[7]

Finally, we define the total training objective:

$$\mathcal{L} = \lambda_{\text{LM}}\mathcal{L}_{\text{LM}} + \lambda_{\text{JSD}}\mathcal{L}_{\text{JSD}} + \lambda_{\text{Margin}}\mathcal{L}_{\text{Margin}} + \lambda_{\text{Sum}}\mathcal{L}_{\text{Sum}},$$

where λ_{LM}, λ_{JSD}, λ_{Margin}, and λ_{Sum} are hyperparameters, all set to 1 for simplicity.

5 Experimental Settings

Models We employ three open-weight LLMs: LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024), MISTRAL-7B-INSTRUCT-V0.3 (Jiang et al., 2023), and GEMMA-2-9B-IT (Team et al., 2024).

[7] Refer to Appendix C for training details.

Model	Method	TriviaQA				MMLU				LogiQA			
		ECE (↓)	|NCE| (↓)	BS (↓)	AUROC (↑)	ECE	|NCE|	BS	AUROC	ECE	|NCE|	BS	AUROC
LLAMA3.1 8B INSTRUCT	Default	16.9	16.6	21.2	56.2	26.9	26.7	29.7	50.8	53.8	53.3	52.9	50.5
	Prompting	12.1	12.0	17.6	59.7	25.5	25.3	28.5	52.8	48.1	47.4	48.0	50.4
	Self-Consistency	15.7$_{\pm 0.1}$	15.1$_{\pm 0.2}$	22.2$_{\pm 0.1}$	58.6$_{\pm 0.9}$	25.0$_{\pm 1.5}$	25.0$_{\pm 1.5}$	29.9$_{\pm 1.1}$	53.9$_{\pm 0.8}$	45.3$_{\pm 1.0}$	45.1$_{\pm 1.1}$	46.2$_{\pm 0.7}$	45.6$_{\pm 1.3}$
	ConfTuner	**5.2**	**1.1**	15.3	66.3	13.9	13.9	24.2	58.2	28.6	28.2	32.5	54.4
	ADVICE (Ours)	10.4$_{\pm 1.2}$	9.8$_{\pm 1.1}$	**14.8**$_{\pm 0.5}$	77.0$_{\pm 0.1}$	**8.6**$_{\pm 2.3}$	7.6$_{\pm 0.7}$	**20.7**$_{\pm 0.7}$	**69.2**$_{\pm 0.9}$	**23.0**$_{\pm 2.7}$	**21.2**$_{\pm 2.9}$	**30.1**$_{\pm 1.9}$	57.9$_{\pm 0.7}$
	w/ ConfTuner	9.4$_{\pm 0.4}$	8.7$_{\pm 0.4}$	15.1$_{\pm 0.1}$	**77.9**$_{\pm 0.1}$	9.6$_{\pm 0.9}$	**7.1**$_{\pm 1.4}$	20.9$_{\pm 0.4}$	68.7$_{\pm 0.8}$	26.0$_{\pm 0.7}$	23.8$_{\pm 0.9}$	31.1$_{\pm 0.4}$	**58.6**$_{\pm 0.3}$
MISTRAL 7B INSTRUCT	Default	32.8	30.5	35.3	51.6	35.3	35.0	37.0	51.5	51.8	50.9	51.3	52.1
	Prompting	27.6	24.8	31.0	52.8	36.3	34.1	38.2	49.9	49.2	48.0	49.2	51.4
	Self-Consistency	31.3$_{\pm 0.4}$	30.3$_{\pm 0.5}$	31.6$_{\pm 0.3}$	68.5$_{\pm 1.0}$	34.5$_{\pm 0.5}$	34.3$_{\pm 0.3}$	36.1$_{\pm 0.3}$	61.0$_{\pm 0.8}$	45.7$_{\pm 0.8}$	45.4$_{\pm 0.6}$	43.2$_{\pm 1.0}$	59.0$_{\pm 2.0}$
	ConfTuner	8.1$_{\pm 1.9}$	7.5$_{\pm 2.3}$	**16.3**$_{\pm 0.5}$	**80.3**$_{\pm 0.4}$	36.0$_{\pm 2.1}$	36.0$_{\pm 2.1}$	34.3$_{\pm 1.3}$	**70.8**$_{\pm 1.6}$	24.8$_{\pm 2.3}$	24.8$_{\pm 2.3}$	**22.1**$_{\pm 2.3}$	63.1$_{\pm 1.1}$
	ADVICE (Ours)	14.5$_{\pm 2.3}$	8.0$_{\pm 2.4}$	20.6$_{\pm 1.2}$	74.7 $_{\pm 2.1}$	28.7$_{\pm 2.5}$	27.3$_{\pm 2.9}$	31.4$_{\pm 1.9}$	62.7$_{\pm 3.1}$	35.6$_{\pm 5.5}$	31.6$_{\pm 6.1}$	38.2$_{\pm 4.3}$	59.3$_{\pm 1.7}$
	w/ ConfTuner	**6.9**$_{\pm 1.2}$	**4.7**$_{\pm 1.4}$	17.8$_{\pm 1.1}$	76.4$_{\pm 1.4}$	**21.8**$_{\pm 0.4}$	**21.3**$_{\pm 0.5}$	**26.5**$_{\pm 0.1}$	69.9$_{\pm 0.3}$	**24.0**$_{\pm 1.0}$	**23.1**$_{\pm 1.1}$	28.1$_{\pm 0.4}$	**68.0**$_{\pm 0.4}$
GEMMA2 9B-IT	Default	21.9	21.8	25.3	52.7	21.0	21.0	24.7	50.1	39.1	39.0	40.4	50.9
	Prompting	21.4	21.3	24.9	53.9	21.0	21.0	24.6	50.4	39.0	38.7	40.3	50.3
	Self-Consistency	28.5$_{\pm 0.5}$	28.2$_{\pm 0.4}$	28.5$_{\pm 0.5}$	65.2$_{\pm 1.1}$	22.5$_{\pm 0.4}$	21.8$_{\pm 0.4}$	25.8$_{\pm 0.2}$	57.4$_{\pm 0.9}$	39.0$_{\pm 0.5}$	38.8$_{\pm 0.5}$	41.8$_{\pm 0.4}$	44.8$_{\pm 0.5}$
	ConfTuner	5.7$_{\pm 0.4}$	2.9$_{\pm 1.5}$	**14.3**$_{\pm 0.3}$	**82.7**$_{\pm 0.2}$	11.0$_{\pm 1.0}$	7.3$_{\pm 0.9}$	20.2$_{\pm 0.3}$	**75.7**$_{\pm 0.4}$	18.4$_{\pm 1.1}$	17.9$_{\pm 1.1}$	**23.8**$_{\pm 0.4}$	**71.4**$_{\pm 0.3}$
	ADVICE (Ours)	6.2$_{\pm 3.2}$	5.1$_{\pm 3.8}$	16.3$_{\pm 0.5}$	77.4$_{\pm 0.5}$	**5.6**$_{\pm 0.4}$	**3.9**$_{\pm 1.0}$	**18.6**$_{\pm 0.3}$	65.9$_{\pm 1.4}$	11.9$_{\pm 2.6}$	10.9$_{\pm 2.8}$	25.8$_{\pm 0.3}$	57.8$_{\pm 0.2}$
	w/ ConfTuner	**3.4**$_{\pm 0.6}$	**2.0**$_{\pm 1.2}$	16.0$_{\pm 0.2}$	77.1$_{\pm 1.0}$	11.7$_{\pm 2.6}$	11.7$_{\pm 2.6}$	19.9$_{\pm 1.0}$	66.8$_{\pm 1.9}$	**8.0**$_{\pm 0.3}$	**6.2**$_{\pm 0.2}$	24.6$_{\pm 0.1}$	69.0$_{\pm 0.3}$

Table 1: Average performance across two seen verbalization types (i.e., Score{Letter, Number}) and three random seeds, with standard deviations. Evaluation is conducted on in-domain (TriviaQA) and out-of-distribution (MMLU and LogiQA) datasets. Best results are in **bold**, and second-best results are underlined. All values are reported as percentages. ADVICE matches ConfTuner and yields orthogonal gains when combined.

Datasets We conduct experiments across three open-ended QA datasets: TriviaQA (Joshi et al., 2017), MMLU (Hendrycks et al., 2021), and LogiQA (Liu et al., 2021). Notably, we train only on TriviaQA, enabling evaluation of out-of-distribution (OOD) generalization.

Confidence verbalization types Following Yang et al. (2025), we adopt five verbalization types:

- **ScoreLetter**: letter grades ({`E`, `D`, `C`, `B`, `A`}).
- **ScoreNumber**: integer scores ({`0`, `1`, . . . , `9`}).
- **ScoreText**: categories ({`low`, `medium`, `high`}).
- **ScoreFloat**: floating-point values ({`0.0`–`1.0`}).
- **ScorePercent**: percentages ({`0`, `1`, . . . , `100`}).

Confidence expressions are ordered in ascending magnitude, with later ones denoting higher confidence. During training, ADVICE uses **ScoreLetter** and **ScoreNumber**; at inference, all scoring types are employed as required by the specific evaluation. More details are provided in Appendix D.

Baselines We also compare against several confidence estimation methods: (1) **Default**, which refers to the naïve use of LLMs with minimal prompting; (2) **Prompting**, which augments the Default approach with explicit instructions to consider the self-generated answer; (3) **Self-Consistency** (Xiong et al., 2024), a sampling-based approach that generates multiple verbalized confidence scores and aggregates them; and (4) **ConfTuner** (Li et al., 2025), which fine-tunes LLMs to align confidence distributions with empirical correctness by optimizing a tokenized Brier score.

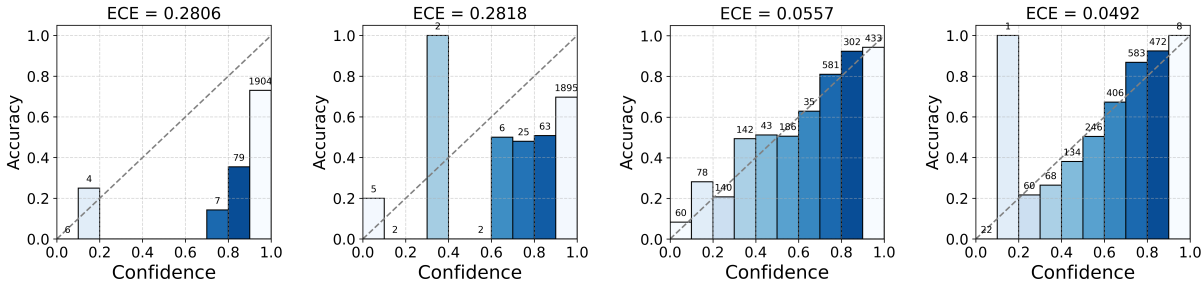
Metrics We evaluate confidence calibration quality with 4 metrics: Expected Calibration Error (**ECE**) (Pakdaman Naeini et al., 2015), absolute Net Calibration Error (**|NCE|**) (Groot and Valdenegro Toro, 2024), Brier score (**BS**) (Brier, 1950), and Area Under the ROC Curve (**AUROC**) (Boyd et al., 2013). Lower values are better for the first three metrics, while higher values are better for AUROC. Appendix A provides further details.

6 Experimental Results

6.1 Main Results

Table 1 summarizes our main experimental results, from which we derive three key findings.

ADVICE leads to improved confidence calibration with strong OOD generalization. When evaluated on TriviaQA, which is used for both training and evaluation, ADVICE consistently outperforms the Default, Prompting, and Self-Consistency baselines, effectively mitigating LLM overconfidence. Compared to ConfTuner, ADVICE achieves comparable performance across diverse settings, supporting the viability of training-based approaches. Beyond aggregate metrics, Figure 5 provides qualitative evidence of this advantage: while Prompting and Self-Consistency pro-



(a) Prompting (b) Self-Consistency (c) ConfTuner (d) ADVICE

Figure 5: Reliability diagrams on TriviaQA with GEMMA-2-9B-IT under the ScoreNumber setting, where numbers above each bin indicate the number of data instances. ADVICE achieves high calibration quality comparable to ConfTuner, demonstrating their effectiveness. Additional cases are illustrated in Figure 14, 15 in the Appendix.

duce uniformly high confidence scores with limited reliability, ADVICE yields fine-grained confidence estimates that more closely track accuracy, resulting in more precise predictions.

Furthermore, as training inherently carries a risk of overfitting, robustness to domain shift is essential for training-based methods. We therefore evaluate ADVICE and ConfTuner under OOD settings using MMLU and LogiQA. ADVICE outperforms ConfTuner in 19 of 24 cases (2 datasets × 4 metrics × 3 models), demonstrating strong robustness as a general-purpose confidence calibration framework.

ADVICE provides orthogonal gains over ConfTuner. Table 1 further shows that combining ADVICE with ConfTuner (i.e., *w/* ConfTuner) often yields extra gains, implying that the two methods tackle orthogonal aspects of the limitations in verbalized confidence estimation. ConfTuner directly aligns confidence token generation with answer predictions using existing datasets, which may make it susceptible to overfitting. In contrast, ADVICE, while also based on fine-tuning, guides the model to condition confidence estimation more strongly on its own generated answers, indirectly improving confidence calibration. As a result, ADVICE offers greater robustness under distribution shift, complementing ConfTuner's shortcomings. Appendix C provides details of combining two methods.

ADVICE achieves the best trade-off between performance and efficiency. Figure 6 visualizes the performance–efficiency relationship for confidence estimation, where performance is measured by ECE and efficiency by the number of generated tokens (i.e., token usage).[8] For both metrics, lower

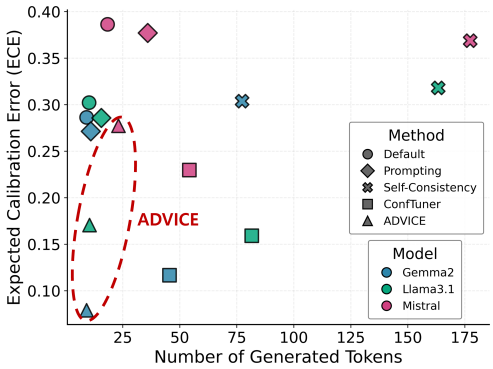


Figure 6: Calibration performance (ECE)–efficiency (token usage) across method–model pairs. Compared to the baselines, ADVICE achieves the best balance, requiring the lowest generation cost while maintaining reliable confidence estimation.

values are better, making the lower-left region ideal. Each point denotes performance averaged over diverse prompt types, datasets, and random seeds.

Across all three LLMs, Default and Prompting are relatively cost-efficient but consistently exhibit high ECE, indicating limited calibration quality. Self-Consistency dramatically increases token usage—reflecting its multi-sampling nature—yet still delivers unsatisfactory performance. On the other hand, ConfTuner improves calibration over these decoding-based baselines but typically incurs higher token usage for confidence verbalization. Notably, ADVICE clusters in the lower-left region, achieving lower ECE with fewer generated tokens and thus offering a more favorable balance between confidence calibration performance and efficiency.

6.2 Ablation Study on Training Objectives

We conduct an ablation study on the final training objective to assess the contribution of its components. First, as described in §4.2, $\mathcal{L}_{\text{LM}}$ serves as an

[8] Token usage is computed as the total number of tokens generated for the answer and the verbalized confidence.

Model	Training Obj.	TriviaQA			MMLU		
		ECE	|NCE|	BS	ECE	|NCE|	BS
GEMMA2 9B-IT	LM	23.0	23.0	25.6	22.5	22.5	25.2
	LM+JSD	8.6	1.1	16.7	13.2	11.7	20.5
	LM+Margin	16.8	**0.9**	20.2	21.9	13.1	24.7
	LM+Sum	21.1	21.1	24.2	19.5	19.5	23.9
	LM+JSD+Margin	11.0	4.0	17.3	14.3	11.3	21.0
	LM+JSD+Sum	15.3	14.8	18.9	7.5	7.1	19.0
	LM+Margin+Sum	20.9	20.2	23.7	19.7	19.7	23.8
	ADVICE	**6.2**	5.1	**16.3**	**5.6**	**0.4**	**18.6**
LLAMA3.1 8B INSTRUCT	LM	13.3	13.3	17.3	27.7	27.7	29.9
	LM+JSD	**6.2**	**3.4**	**14.2**	20.5	20.1	25.1
	LM+Margin	17.4	6.8	20.6	28.2	24.6	30.2
	LM+Sum	32.5	32.5	26.8	20.3	20.3	26.0
	LM+JSD+Margin	11.0	7.4	16.8	23.3	19.3	26.4
	LM+JSD+Sum	19.5	19.5	17.8	**5.5**	**2.0**	**19.8**
	LM+Margin+Sum	31.0	31.0	25.3	15.1	15.1	23.0
	ADVICE	10.4	9.8	14.8	8.6	7.6	20.7

Table 2: Ablation study of the final training objective. All values are reported as percentages. Best and second-best results are indicated in **bold** and underlined.

Model	Verb. Type	Method	TriviaQA			MMLU		
			ECE	|NCE|	BS	ECE	|NCE|	BS
GEMMA2 9B-IT	Text	Default	17.1	17.1	22.3	25.5	25.5	26.5
		ADVICE	**8.3**	**6.7**	**17.6**	**6.8**	**0.9**	**18.4**
	Percent	Default	26.9	26.8	27.0	25.5	25.5	26.5
		ADVICE	**6.7**	**0.9**	**16.4**	**8.9**	**8.7**	**19.2**
	Float	Default	27.5	27.4	27.3	26.3	26.3	27.0
		ADVICE	**6.2**	**4.9**	**15.4**	**9.2**	**8.7**	**19.2**
LLAMA3.1 8B INSTRUCT	Text	Default	26.9	26.8	27.0	25.5	25.5	26.5
		ADVICE	**11.9**	**11.9**	**16.4**	**8.0**	**3.7**	**21.5**
	Percent	Default	18.6	18.5	20.3	30.2	30.2	31.6
		ADVICE	**5.3**	**4.1**	**14.0**	**13.2**	**12.9**	**22.7**
	Float	Default	19.6	19.5	21.0	32.5	32.3	33.0
		ADVICE	**5.7**	**5.7**	**14.1**	**14.9**	**14.7**	**22.3**

Table 3: Performance on unseen verbalization formats: ScoreText, ScorePercent, and ScoreFloat. All values are percentages. The best results are in **bold**.

Dataset	Method	GEMMA2-9B-IT		LLAMA3.1-8B-INSTRUCT	
		ScoreLetter	ScoreNumber	ScoreLetter	ScoreNumber
TriviaQA	Default	70.7	70.7	75.2	74.8
	ADVICE	$71.6_{\pm 0.2}$	$71.7_{\pm 0.4}$	$78.1_{\pm 0.1}$	$77.4_{\pm 0.5}$
MMLU	Default	72.2	72.1	66.6	67.2
	ADVICE	$73.0_{\pm 0.1}$	$72.8_{\pm 0.3}$	$67.1_{\pm 0.3}$	$67.5_{\pm 0.76}$
LogiQA	Default	53.2	52.3	39.0	38.9
	ADVICE	$52.8_{\pm 0.2}$	$52.5_{\pm 0.1}$	$41.9_{\pm 0.2}$	$41.6_{\pm 0.1}$

Table 4: Task (QA) accuracies before and after fine-tuning. These results demonstrate that ADVICE does not adversely impact the task performance of LLMs.

auxiliary term for preserving language modeling performance. As a result, we observe that $\mathcal{L}_{\text{LM}}$ is generally independent of confidence calibration. Second, the results in Table 2 show that $\mathcal{L}_{\text{JSD}}$ and $\mathcal{L}_{\text{Margin}}$ contribute to improving confidence verbalization together. Compared to using either loss alone, jointly optimizing both enables the model not only to distinguish P_{correct} from P_{wrong} but also to separate them in the intended direction. Finally, $\mathcal{L}_{\text{Sum}}$ enforces the definition of confidence, resulting in improved calibration on OOD benchmarks. Overall, ADVICE combines these terms to achieve robust generalization across datasets and models.

6.3 Generalization on Verbalization Types

As explained in §5, we construct the training set for ADVICE using two verbalization types (ScoreLetter and ScoreNumber). We then evaluate ADVICE on three additional formats—ScoreText, ScoreFloat, and ScorePercent—to validate its robustness across verbalization schemes. Table 3 reports calibration performance on these unseen types, demonstrating consistent performance across formats.

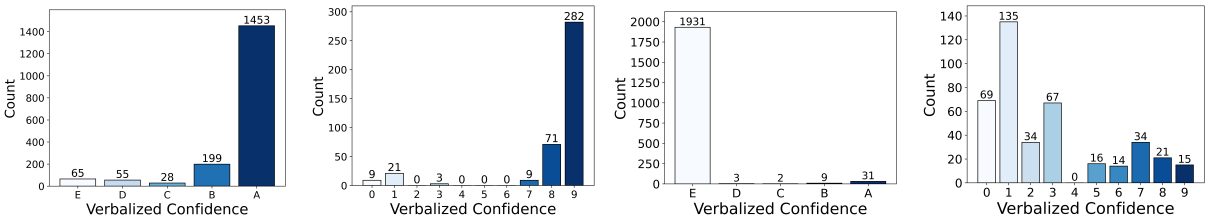
6.4 Effect on General Task Performance

When fine-tuning an LLM, it is essential to verify that the modification does not compromise general task performance. Accordingly, we examine the impact of ADVICE on task (QA) accuracy. As shown in Table 4, accuracy changes are negligible, meaning that ADVICE preserves the LLM's capabilities. Since verbalized confidence is often overconfident, increases in task accuracy could affect calibration metrics such as ECE even if confidence estimation remains unchanged. However, stable accuracy after fine-tuning suggests that the observed ECE reductions stem from improved confidence calibration rather than gains in task performance.

7 ADVICE Enhances Answer Awareness

Ultimately, we conduct analyses to provide evidence that ADVICE's improvements originate from its answer-groundedness. To this end, we introduce a new experiment and revisit the analyses presented in §3 after training LLMs with ADVICE.

In the first study, we replace answer tokens with an equal-length sequence of padding (e.g., <pad>) tokens to simulate the absence of the answer and evaluate its effect. As illustrated in Figures 7a and 7b, the Default method (i.e., before training) produces confidence distributions that are markedly skewed toward high values, indicating overconfidence. In contrast, ADVICE (Figures 7c and 7d; after training) reveals the opposite behavior: its verbalized confidence substantially declines when the answer is masked, conveying obscurity regarding the correctness of the response. This finding empir-

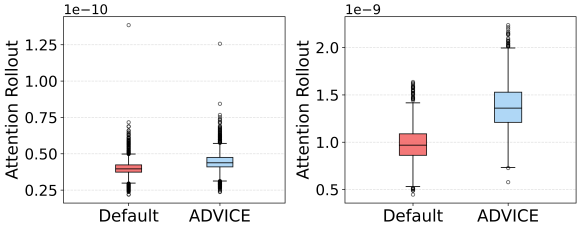


(a) LLAMA3.1 (TriviaQA, Default) (b) MISTRAL (LogiQA, Default) (c) LLAMA3.1 (TriviaQA, ADVICE) (d) MISTRAL (LogiQA, ADVICE)

Figure 7: Verbalized confidence distributions after answer masking. Default remains overconfident without answers, whereas ADVICE reallocates probability mass toward less confident expressions (e.g., E, 0, and 1).

Training Step (ADVICE)	**Top 10 Tokens** 1	2	3	4	5	6	7	8	9	10
0 (Default)	<start_of_turn>	<bos>	user	_only	Provide	Confidence	Answer	\n	“‘	Answer
100	user	<start_of_turn>	Provide	_only	\n	\n	<bos>	Confidence	Answer	Answer
200	<start_of_turn>	<bos>	user	Provide	_only	\n	_A	**_Exile**	Confidence	_between
300	<start_of_turn>	**_Exile**	user	<bos>	_only	Provide	_Kiss	Confidence	_You	Question
400	**_Exile**	<start_of_turn>	_only	<bos>	Provide	user	_A	Confidence	Question	“‘
500	<start_of_turn>	<bos>	**_Exile**	_only	user	Provide	_A	Confidence	“‘	Question

Table 5: Top 10 tokens sorted by their absolute attribution scores for GEMMA2-9B-IT. We observe an increase in the rank of the answer token (**`_Exile`**), suggesting that ADVICE promotes greater answer dependence in the model.



(a) GEMMA2-9B-IT (b) LLAMA3.1-8B-INSTRUCT

Figure 8: Attention Rollout score distributions for Confidence (C) $\rightarrow$ Answer (A), comparing ADVICE and Default. ADVICE contributes to improved attention. In both cases, the t-test confirms statistically significance.

ically validates that ADVICE enhances the model's answer-awareness in confidence estimation.

Second, we revisit the Attention Rollout analysis (§3.2) to examine how adopting ADVICE alters the attention dynamics compared to the Default method (Default vs. ADVICE). Figure 8 illustrates that compared to Default, ADVICE consistently directs the model's attention more strongly toward the answer. These results support our hypothesis that poor confidence verbalization arises from answer-independence, and that ADVICE improves performance by mitigating this limitation.

Finally, we conduct a qualitative analysis of token attribution scores using Integrated Gradients, following the same procedure as in §3.2. Using a fixed input (*Instruction*, Q, A), we track how token-level attribution patterns evolve throughout the fine-tuning process of ADVICE. Specifically, we focus on the top-k tokens ($k = 10$) ranked by the absolute magnitude of their attribution scores, capturing both positive and negative contributions. In Table 5, we can see that as training progresses, the rank of the answer token (`_Exile`) increases, suggesting that ADVICE encourages the model to become more answer-dependent.

To summarize, our three experiments in this section consistently demonstrate that LLMs' overconfidence mainly arises from neglecting answer information in verbalized confidence estimation, and that ADVICE effectively mitigates this problem, by solving the primary cause of overconfidence in LLMs.

8 Conclusion

This work provides a systematic investigation into the fundamental cause of overconfidence in LLMs' verbalized confidence. In particular, our mathematical analysis identifies *answer-independence* as the key contributing factor. Based on this insight, we propose **ADVICE** (**A**nswer-**D**ependent **V**erbal**I**zed **C**onfidence **E**stimation), an intuitive and effective training framework that guides LLMs to generate more answer-grounded confidence estimations. Extensive experiments demonstrate that ADVICE substantially mitigates the overconfidence commonly observed in LLMs, enabling them to produce more reliable and better-calibrated confidence estimates.

Limitations

This study identifies the primary cause of overconfidence in LLMs and presents ADVICE, which effectively addresses it, leading to notable improvements in calibration. However, several limitations remain, offering directions for future research.

First, while ADVICE enhances calibration through a contrastive objective that promotes answer-dependent confidence, it requires LLM-generated answers to form contrastive pairs, introducing additional data construction costs. Nevertheless, we consider this trade-off reasonable, as it explicitly targets the fundamental factor behind overconfidence and advances the development of more reliable models.

Second, this work primarily focuses on short-form QA and multiple-choice question answering. Extending the approach to tasks that demand long-context understanding and complex reasoning would be a valuable next step.

Third, calibration performance is inherently coupled with task accuracy: in high-accuracy regimes, even the Default method can appear well calibrated. For example, on SciQ—where models achieve over 90% accuracy—the base model attains the best calibration performance (see Figure 6 in the Appendix). We observe a similar pattern across other confidence estimation methods, highlighting the need for more rigorous evaluation practices in this literature.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Kendrick Boyd, Kevin H. Eng, and C. David Page. 2013. Area under the precision-recall curve: Point estimates and confidence intervals. In *Machine Learning and Knowledge Discovery in Databases*, pages 451–466, Berlin, Heidelberg. Springer Berlin Heidelberg.

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

Oscar Blessed Deho, Michael Bewong, Selasi Kwashie, Jiuyong Li, Jixue Liu, Lin Liu, and Srecko Joksimovic. 2025. Is it still fair? a comparative evaluation of fairness algorithms through the lens of covariate drift. *Mach. Learn.*, 114(1).

Kobe Desender, K Richard Ridderinkhof, and Peter R Murphy. 2021. Understanding neural signals of post-decisional performance monitoring: An integrative review. *eLife*, 10:e67556.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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 *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 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*.

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

Thanmay Jayakumar, Fauzan Farooqui, and Luqman Farooqui. 2023. Large language models are legal but they are not: Making the case for a powerful LegalLLM. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 223–229, Singapore. 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. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

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.

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

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *Preprint*, arXiv:2509.04664.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. 2025. Taming overconfidence in LLMs: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations*.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. 2025. Conftuner: Training large language models to express their confidence verbally. *Preprint*, arXiv:2508.18847.

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

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

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2021. Logiqa: a challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

M.L. Menéndez, J.A. Pardo, L. Pardo, and M.C. Pardo. 1997. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318.

Aiswariya Milan Kummaya, Amudha Joseph, Kumar Rajamani, and George Ghinea. 2025. Fed-hetero: A self-evaluating federated learning framework for data heterogeneity. *Applied System Innovation*, 8(2).

Hadi Mohammadi, Ayoub Bagheri, Anastasia Giachanou, and Daniel L. Oberski. 2025. Explainability in practice: A survey of explainable nlp across various domains. *Preprint*, arXiv:2502.00837.

Joaquin Navajas, Bahador Bahrami, and Peter E Latham. 2016. Post-decisional accounts of biases in confidence. *Current Opinion in Behavioral Sciences*, 11:55–60. Computational modeling.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

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

Hajar Sakai and Sarah S. Lam. 2025. Large language models for healthcare text classification: A systematic review. *Preprint*, arXiv:2503.01159.

Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. 2025. Rewarding doubt: A reinforcement learning approach to confidence calibration of large language models. *CoRR*, abs/2503.02623.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

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.

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

Chenjun Xu, Bingbing Wen, Bin Han, Robert Wolfe, Lucy Lu Wang, and Bill Howe. 2025. Do language models mirror human confidence? exploring psychological insights to address overconfidence in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25655–25672, Vienna, Austria. Association for Computational Linguistics.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. 2025. On verbalized confidence scores for LLMs. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. 2024. Fact-and-reflection (FaR) improves confidence calibration of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8702–8718, Bangkok, Thailand. Association for Computational Linguistics.

Ziang Zhou, Tianyuan Jin, Jieming Shi, and Qing Li. 2025. Steerconf: Steering llms for confidence elicitation. *Preprint*, arXiv:2503.02863.

A Experimental Setting Description

Here we provide the detailed settings for the experiments described in §3.1 and §5.

First, by comparing confidence distributions in §3.1, we further demonstrate the answer independence of verbalized confidence. For evaluating answer-independence, we leverage the training set of TriviaQA. We construct multiple answers corresponding to the same question in the dataset, i.e., $(q, \{a_1, \ldots, a_m\})$. We set $m = 30$ for TriviaQA to reflect its free-form generation setting, whereas for MCQ datasets we set $m = 4$ by leveraging the pre-defined distractors provided with each dataset. We then remove duplicate answers to construct the $\hat{A}_q = \{a_1, \ldots, a_n\}$ for each question in dataset. Note that the number of filtered answers, n, depends on the question q. Furthermore, it is important to emphasize that the answer candidates a_i and a_j are selected based on their distinct informational content, rather than their ground-truth correctness. The objective of this analysis is to demonstrate that even when the model is presented with two different answers containing different information, the verbalized confidence remains nearly identical.

Second, we also provide detailed explanations for metrics used in §5. ECE is defined as follows:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|,$$

where M denotes the number of bins, N the total number of samples, B_m the collection of instances assigned to the m-th bin, acc the accuracy, and conf the confidence. We set $M = 10$, a value commonly used in practice. ECE quantifies the average absolute difference between predicted confidence and empirical accuracy over grouped confidence intervals.

We also employ NCE, a variation of ECE, to complement each other. We modify NCE by taking its absolute value for more intuitive interpretation, so that smaller value indicates better calibration.

|NCE| is formulated as:

$$|\mathrm{NCE}| = |\sum_{m=1}^{M} \frac{|B_m|}{N} (\mathrm{acc}(B_m) - \mathrm{conf}(B_m))|.$$

The distinction is that NCE computes a weighted sum of signed differences, whereas ECE computes one of absolute differences. As a result, biased confidence estimation, such as over- or under-confidence, yields a large absolute NCE value.

The Brier score is defined as the mean squared difference between predicted confidence scores (c_n) and true binary outcomes (y_n), directly measuring the accuracy of probabilistic predictions. It is calculated as:

$$\mathrm{BS} = \frac{1}{N} \sum_{n=1}^{N} (y_n - c_n)^2,$$

Finally, AUROC measures the likelihood that a randomly selected positive instance receives a higher confidence score than a randomly selected negative one, reflecting the model's overall ability to rank predictions by confidence.

B LLM Probing Methods

Attention Rollout Compared to naive attention scores, Attention Rollout provides more reliable attributions by recursively aggregating attention across layers. This aggregation accounts for the residual connections and the hierarchical flow of information, yielding a more faithful estimate of token contributions.

Attention Rollout is recursively defined as:

$$\tilde{A}(l_i) = \begin{cases} A(l_i)\tilde{A}(l_{i-1}), & \text{if } i > j, \\ A(l_i), & \text{if } i = j, \end{cases}$$

where $A(l_i)$ denotes the raw attention matrix of layer i, updated with residual connections and computed as

$$A(l_i) = 0.5\, W_{\text{att},i} + 0.5I.$$

We define the Question tokens as those ranging from `Question:` to the end of the input (i.e., the `<end_of_turn>` token), and the Answer tokens as those spanning from `Answer:` to the token immediately preceding the subsequent `Confidence:`. As the next step, we computed the attention rollout for each token, starting from the position of the colon (i.e., the ":" in `Answer:`), which corresponds to the point where the first token of the answer or the confidence expression begins to be generated. Subsequently, we compute the rollout scores across the entire layer, and aggregate the rollout scores by taking their average.

Integrated Gradients Integrated Gradients are formulated as:

$$(x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

where i denotes the feature dimension, and x'_i corresponds to the baseline input. In practice, the integral is approximated via a Riemann sum with a predefined number of interpolation steps, n_steps. We employed the `IntegratedGradients` implementation from the `captum` library to compute attribution scores. For all experiments, we set the hyperparameters to $n_steps = 1024$ and $internal_batch_size = 32$, and adopted a zero vector as the baseline. Furthermore, we visualized the resulting attributions using the visualization utilities provided within the same package.

C Implementation Details

Training Details Here, we describe the implementation details of ADVICE. We utilize LoRA (Hu et al., 2022) from the HuggingFace PEFT library (Mangrulkar et al., 2022) for fine-tuning.

Specifically, we fine-tune the adapters attached to the query, key, value, and output projection modules across all transformer layers, using a rank of $r = 16$ and a scaling factor of $\alpha = 32$. Optimization is performed with AdamW at a learning rate η, learning rate warmup over the first 5% steps, and linear decay of the learning rate ($\eta = 3 \times 10^{-5}$ for MISTRAL-7B-INSTRUCT-V0.3, $\eta = 1 \times 10^{-5}$ for the rest). We adopt a batch size of 16 and apply gradient accumulation with a factor of 2 using the Accelerate framework. We train all LLMs for 4 epochs. All training runs are conducted on 1 NVIDIA H200 NVL PCIe GPU.

Based on the results in Figure 9, we set δ_{JSD} to 0.6, as the Jensen–Shannon divergence (JSD) is defined to take values between 0 and $\ln 2$ (≈ 0.693). The value of δ_{Margin} is set to 1, which is chosen to be strictly greater than the expectation difference observed in all experimental settings. Explanation for each training objective hyperparameter is described in §4.2.

Note that the size of our training dataset varies depending on the generated texts of each LLM, as our dataset construction process leverages the model itself to generate samples using stochastic decoding. Consequently, we obtain nearly 1k, 1k, and 2k training samples for GEMMA-2-9B-IT, LLAMA-3.1-8B-INSTRUCT, and MISTRAL-7B-INSTRUCT-V0.3, respectively.

For optimization stability, we impose within-batch homogeneity: although training employs multiple verbalization variants, each mini-batch contains a single verbalization type, and batches are shuffled across steps.

Self-Consistency Following Xiong et al. (2024), we implement the method using the vanilla prompt with $M = 5$. Specifically, we select the Avg-Conf variant of the method, which computes the weighted sum of confidence scores and this configuration has been shown to outperform other ones. This involves prompting the LLM to generate five candidate answers and aggregating them as follows:

$$C_{\text{conf}} = \frac{\sum_{i=1}^{M} \mathcal{I}\{\hat{Y}_i = \tilde{Y}\} \times C_i}{\sum_{i=1}^{M} C_i},$$

where $\hat{Y}_i$ are candidate answers with their corresponding verbalized confidence C_i and $\mathcal{I}$ is indicator function. Note that $\tilde{Y}$ denotes the answer that has the highest confidence score among all candidate answers.

ConfTuner We re-implement ConfTuner based on their official code.[9] For LLAMA-3.1-8B-INSTRUCT, we use their publicly available fine-tuned model.[10] We fine-tune GEMMA-2-9B-IT and MISTRAL-7B-INSTRUCT-V0.3 on our training dataset. Following the original implementation, we adopt the same prompt type (i.e., **ScoreNumber**). Since the number of training samples differs from the original paper, we also adjust the number of training epochs accordingly: we fine-tune MISTRAL-7B-INSTRUCT-V0.3 for 3 epochs and GEMMA-2-9B-IT for 2 epochs.

ADVICE *w/* ConfTuner We integrate our training objective with ConfTuner's calibration loss. Specifically, using the correct and incorrect answers in our training dataset, we adapt ConfTuner's calibration loss to our setting and re-define $\mathcal{L}\text{cal}$ accordingly. We then train three models with five objectives, i.e., $\mathcal{L}_{\text{LM}}, \mathcal{L}_{\text{JSD}}, \mathcal{L}_{\text{Margin}}, \mathcal{L}_{\text{Sum}}, \mathcal{L}_{\text{cal}}$, following the definition of $\mathcal{L}_{\text{cal}}$ in Li et al. (2025). We set λ_{LM}, λ_{JSD}, λ_{Margin}, and λ_{Sum} to 0.5, and all coefficients of $\mathcal{L}_{\text{cal}}$ to 1. We fine-tune GEMMA-2-9B-IT for 3 epochs and MISTRAL-7B-INSTRUCT-V0.3 for 4 epochs. We optimize with AdamW at a learning rate $\eta = 3 \times 10^{-5}$. Moreover, we further train their fine-tuned model using the loss described in §4.2 for LLAMA-3.1-8B-INSTRUCT.

[9] https://github.com/liushiliushi/ConfTuner
[10] liushiliushi/ConfTuner-LLaMA

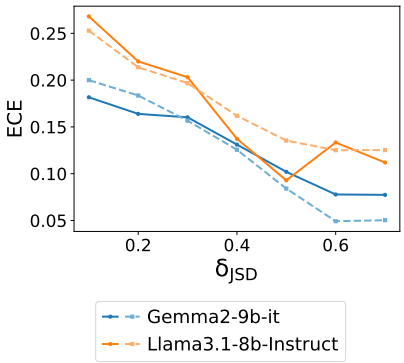


Figure 9: ECE as a function of δ_{JSD} on TriviaQA. Blue lines correspond to GEMMA-2-9B-IT, and orange lines to LLAMA-3.1-8B-INSTRUCT.

Model	Method	SciQ			
		ECE	|NCE|	BS	AUROC
GEMMA2 9B-IT	Default	4.9	1.5	**5.2**	50.3
	Prompting	5.1	**1.4**	**5.2**	50.0
	Self-Consistency	**4.0**$_{\pm 0.1}$	4.0$_{\pm 0.1}$	6.3$_{\pm 0.1}$	70.3$_{\pm 0.6}$
	ConfTuner	6.8$_{\pm 0.7}$	2.9$_{\pm 0.8}$	6.9$_{\pm 0.4}$	**77.2**$_{\pm 0.5}$
	ADVICE (Ours)	14.2$_{\pm 4.4}$	14.2$_{\pm 4.4}$	7.4$_{\pm 1.5}$	72.6$_{\pm 2.8}$
LLAMA3.1 8B INSTRUCT	Default	**3.7**	2.1	8.2	52.0
	Prompting	4.3	**0.7**	7.2	51.4
	Self-Consistency	**1.5**$_{\pm 0.3}$	1.3$_{\pm 0.2}$	7.1$_{\pm 0.2}$	**74.0**$_{\pm 1.0}$
	ConfTuner	9.5	9.1	9.5	58.4
	ADVICE (Ours)	7.9$_{\pm 2.1}$	7.7$_{\pm 2.2}$	**6.6**$_{\pm 0.3}$	73.4$_{\pm 3.5}$
MISTRAL 7B INSTRUCT	Default	10.3	9.0	15.3	50.3
	Prompting	9.8	8.5	14.9	51.7
	Self-Consistency	14.8$_{\pm 0.4}$	14.5$_{\pm 0.3}$	16.9$_{\pm 0.4}$	67.1$_{\pm 0.7}$
	ConfTuner	10.8$_{\pm 1.1}$	9.9$_{\pm 1.7}$	14.0$_{\pm 0.5}$	**68.6**$_{\pm 0.9}$
	ADVICE (Ours)	**8.3**$_{\pm 0.9}$	**8.1**$_{\pm 1.1}$	**12.5**$_{\pm 0.7}$	63.0$_{\pm 3.8}$

Table 6: Average performance over trained verbalization types (i.e., Score{Letter, Number} for ADVICE), evaluated on SciQ. Values are percentages. Best results are in **bold**—minimum for ECE and BS, absolute minimum for NCE, and maximum for AUROC.

D Confidence Verbalization Types

As outlined in §5, we utilize five types of verbalization—**ScoreLetter**, **ScoreNumber**, **ScoreText**, **ScoreFloat** and **ScorePercent**. We train ADVICE with **ScoreLetter** and **ScoreNumber** and evaluate on (1) the same two prompt types and (2) the other prompt types, where the latter serves as a generalization test in Table 3. Following Li et al. (2025), we train and evaluate LLMs in the **ScoreNumber** setting for ConfTuner. To quantify calibration metrics, each verbalized confidence expression is mapped to a numeric value within the interval $[0, 1]$. We specify the numeric mappings for each prompt type as follows:

- **ScoreLetter**: Each letter token {`E`, `D`, `C`, `B`, `A`} is mapped to: `E`$= 0.1$, `D`$= 0.3$, `C`$= 0.5$, `B`$= 0.7$, `A`$= 0.9$.
- **ScoreNumber**: Each digit $i \in \{\texttt{0}, \texttt{1} \ldots, \texttt{9}\}$ is assigned a value of $i/9$.
- **ScoreText**: Verbalized levels are mapped as `low` $= 0.1$, `medium` $= 0.5$, `high` $= 0.9$.
- **ScoreFloat**: Each floating-point value is used directly without further mapping.
- **ScorePercent**: Each percentage token $i\%$ is mapped to a value of $i/100$.

E Prompt Templates

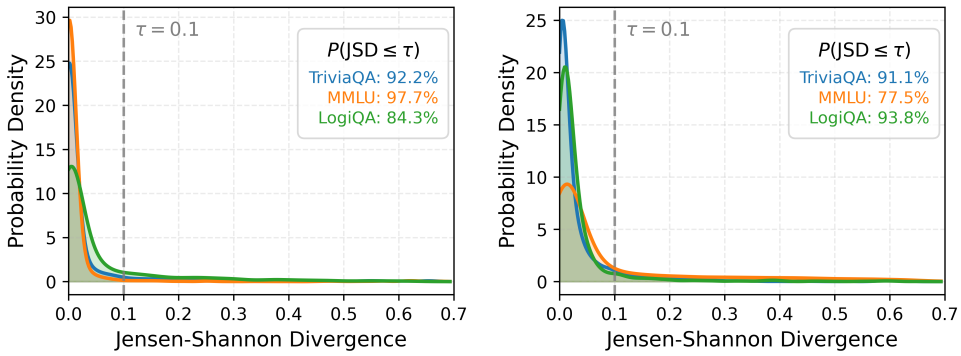
We provide the prompt templates, as shown in Table 8 and Table 9, following the formats used by Yang et al. (2025). For ConfTuner, we use the template in Table 10 proposed by Li et al. (2025).

F Effect of Hyperparameter

To examine the impact of $\mathcal{L}_{\text{JSD}}$, we evaluate the calibration performance under varying δ_{JSD}. The hyperparameter δ_{JSD} controls how sensitively the model distinguishes between the two answer distributions, P_{correct} and P_{wrong}. Figure 9 shows the variation of ECE across different values of δ_{JSD}. We consistently observe a reduction in ECE as δ_{JSD} increases. This is intuitive, as a smaller δ_{JSD} reduces the penalty for similarity between contrastive distributions, resulting in less distinct separation and degraded calibration performance.

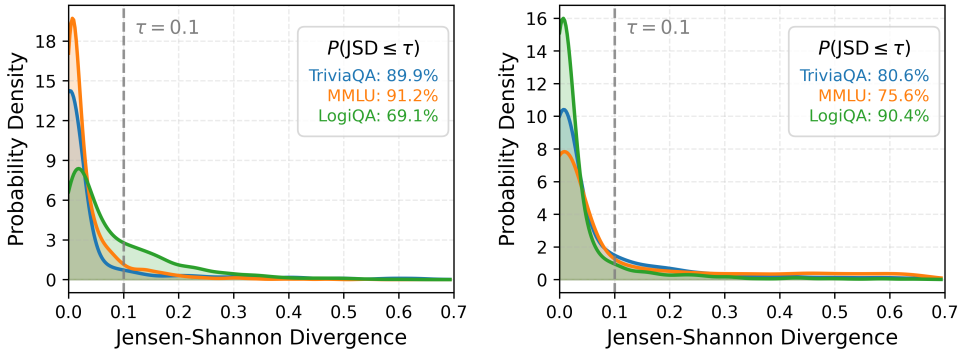
G Qualitative Evaluation

We qualitatively assess how our method affects the extent to which confidence is grounded in the answer. In Figure 16 and Figure 17, we observe that as training progresses, the attribution scores of answer tokens gradually increase. This result demonstrates that our method enhances calibration capability by inducing answer-dependent confidence estimation.



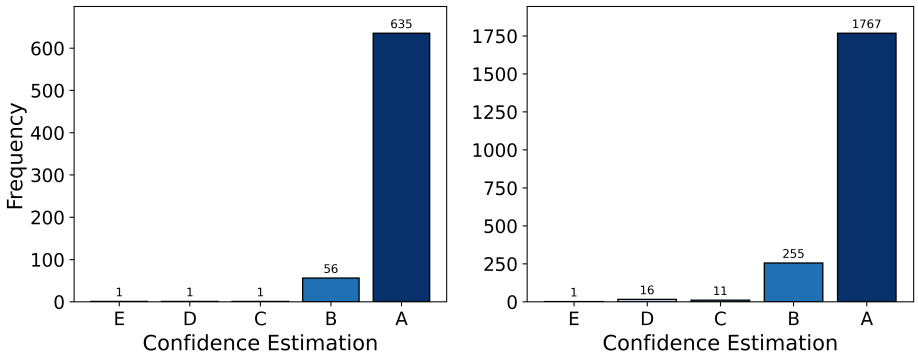
(a) GEMMA2-9B-IT (b) LLAMA3.1-8B-INSTRUCT

Figure 10: Probability density functions (PDFs) of the set $\{\text{JSD}(P_M(C|q,a_i)||P_M(C|q,a_j))\}$, where $M \in \{\text{GEMMA2}, \text{LLAMA3.1}\}$ and (q, a_i, a_j) are from TriviaQA, MMLU, LogiQA under ScoreNumber setting. Each PDF (solid curve) is computed via Gaussian kernel density estimation. Near-zero concentration implies answer-independent confidence.



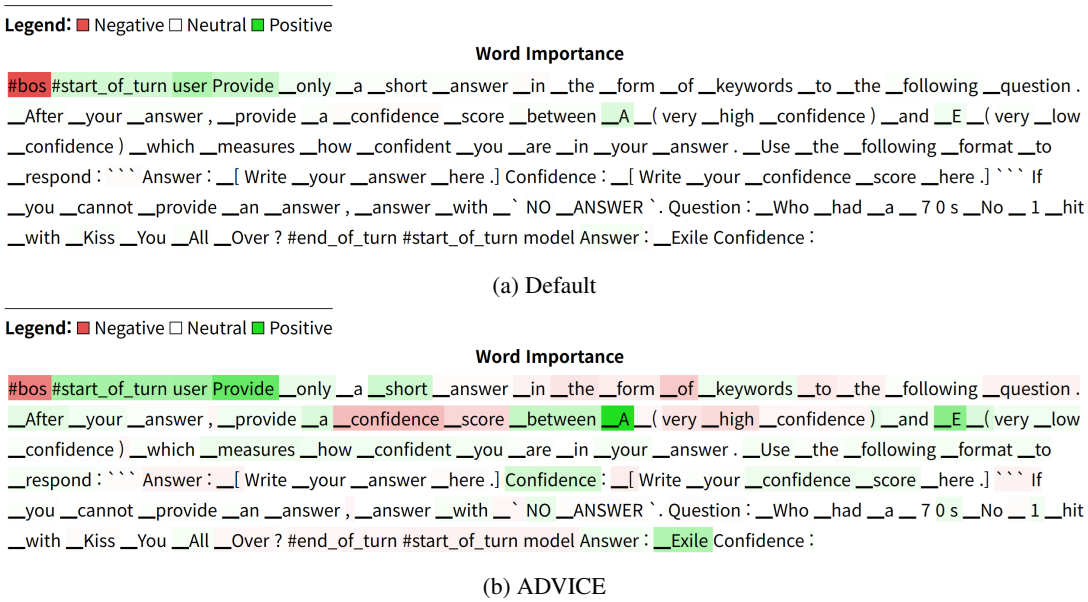
(a) GEMMA2-9B-IT (b) LLAMA3.1-8B-INSTRUCT

Figure 11: Probability density functions (PDFs) of the set $\{\text{JSD}(P_M(C|q,a_i)||P_M(C|q,a_j))\}$, where $M \in \{\text{GEMMA2}, \text{LLAMA3.1}\}$ and (q, a_i, a_j) are from TriviaQA, MMLU, LogiQA under ScoreLetter setting. In this figure, a_i denotes the correct answer for q, while a_j is restricted to one of the sampled wrong answers; all other settings are identical to Figure 2. Consistent with Figure 2, this results reveal little change in the confidence distribution between correct and wrong answers, suggesting that verbalized confidence is largely insensitive to answer correctness.



(a) Correct answer (b) Wrong answer

Figure 12: Histograms of verbalized confidence for correct and wrong answers. Both distributions are skewed toward high confidence, indicating that LLMs often express high confidence regardless of correctness. Together with Figure 2, this motivates comparing different incorrect answers to assess whether verbalized confidence is answer-dependent.



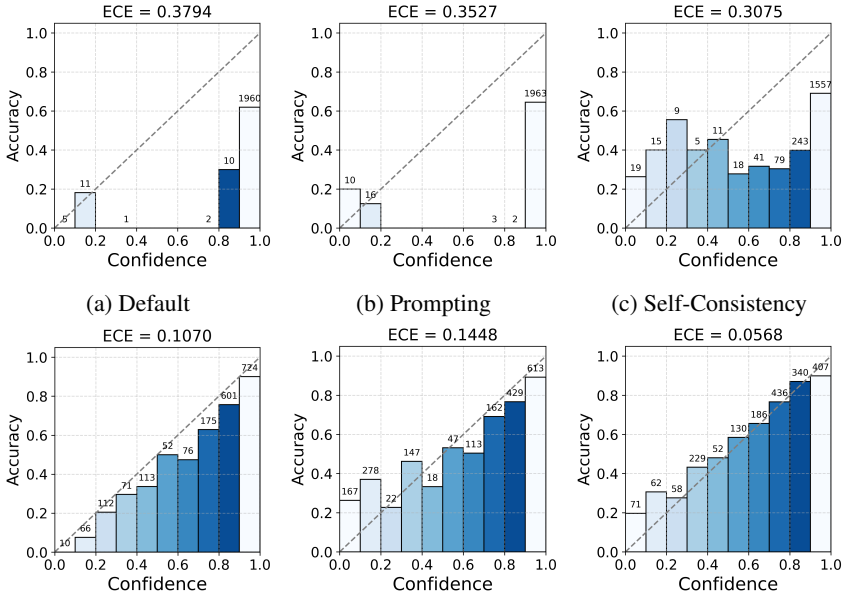
(a) Default

(b) ADVICE

Figure 13: Visualization of token attribution with Integrated Gradients (GEMMA2-9B-IT).

Question	Correct answer (a_{correct})	Wrong answers	Wrong answer (a_{wrong})
Which state renewed Mike Tyson's boxing license in 1998?	Nevada	[Connecticut, Oregon, California]	California
Who were the first team to field an all foreign starting line up in the English Premiership?	Chelsea	[West Ham United, Aston Villa]	West Ham United
On a standard keyboard, which is the largest key?	Space bar	[Shift key, Enter key]	Shift key
Which London railway station has the most platforms?	London Waterloo	[London Victoria, London Paddington, London Liverpool]	London Victoria
If a month has a Friday the thirteenth then on what day of the week would that month begin?	Sunday	[Wednesday, Thursday, Monday, Friday, Tuesday]	Thursday
Which major British newspaper closed down for almost a year in 1978?	The Times	[Daily Mirror, The Sun, News of the World]	The Sun

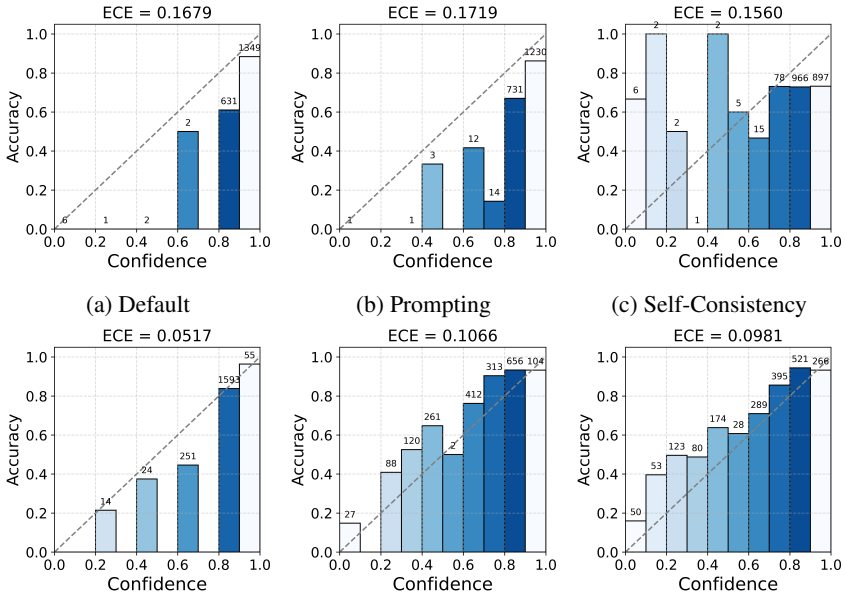
Table 7: Examples of triplet $(q, a_{\text{correct}}, a_{\text{wrong}})$ in our training dataset. Although a_{wrong} is obtained via stochastic decoding, these samples frequently include hard negatives—answers that are semantically plausible and contextually relevant to the question but remain factually incorrect. This diversity encourages the model to learn fine-grained distinctions between correct information and subtly incorrect alternatives.



(a) Default (b) Prompting (c) Self-Consistency

(d) ConfTuner (e) ADVICE (f) ADVICE *w/* ConfTuner

Figure 14: Reliability diagrams of MISTRAL-7B-INSTRUCT-V0.3 on TriviaQA under the ScoreNumber setting, where numbers above each bin indicate the number of instances.



(a) Default (b) Prompting (c) Self-Consistency

(d) ConfTuner (e) ADVICE (f) ADVICE *w/* ConfTuner

Figure 15: Reliability diagrams of LLAMA-3.1-8B-INSTRUCT on TriviaQA under the ScoreNumber setting, where numbers above each bin indicate the number of instances.

Task	**Prompt**
Generation	Provide only a short answer in the form of keywords to the following question.
Multiple-Choice	The following multiple-choice question has only one correct answer. Provide only the option letter of the correct answer.

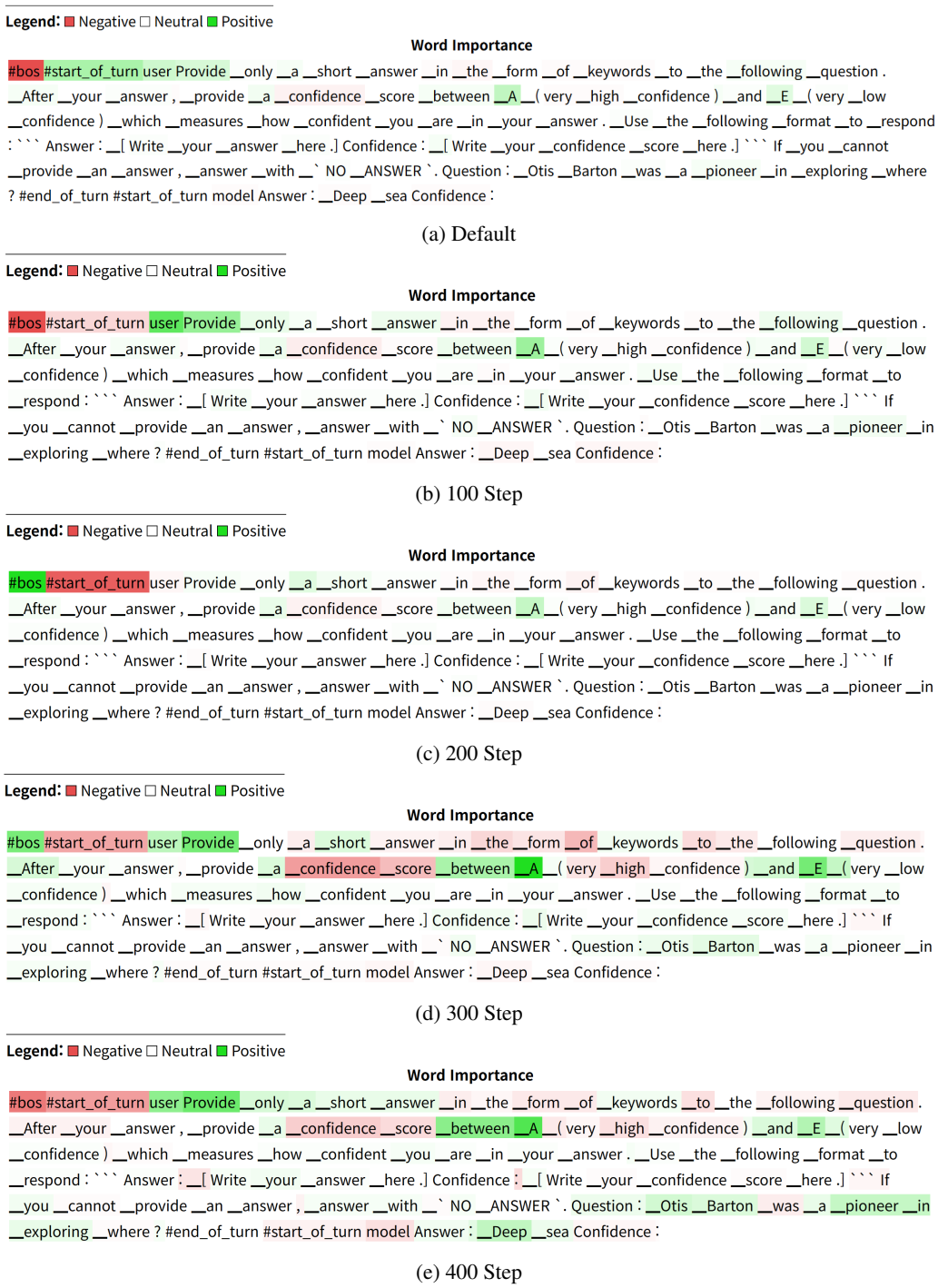
Table 8: These are task-dependent prefix prompts that are placed before the main prompt template.

Type	Prompt
ScoreLetter	After your answer, provide a confidence score between A (very high confidence) and E (very low confidence) which measures how confident you are in your answer. Use the following format to respond: ‘‘‘ Answer: [Write your answer here.] Confidence: [Write your confidence score here.] ‘‘‘ If you cannot provide an answer, answer with ‘NO ANSWER‘.
ScoreNumber	After your answer, provide a confidence score between 0 and 9 which measures how confident you are in your answer, where 9 is the maximum. Never use 10. Use the following format to respond: ‘‘‘ Answer: [Write your answer here.] Confidence: [Write your confidence score here.] ‘‘‘ If you cannot provide an answer, answer with ‘NO ANSWER‘.
ScoreText	After your answer, provide one of the following confidence scores which measures how confident you are in your answer: high, medium, low. Use the following format to respond: ‘‘‘ Answer: [Write your answer here.] Confidence: [Write your confidence score here.] ‘‘‘ If you cannot provide an answer, answer with ‘NO ANSWER‘.
ScorePercent	After your answer, provide a confidence score in percentage which measures how confident you are in your answer. Use the following format to respond: ‘‘‘ Answer: [Write your answer here.] Confidence: [Write your confidence score here.] ‘‘‘ If you cannot provide an answer, answer with ‘NO ANSWER‘.
ScoreFloat	After your answer, provide a confidence score between 0.0 and 1.0 which measures how confident you are in your answer. Use the following format to respond: ‘‘‘ Answer: [Write your answer here.] Confidence: [Write your confidence score here.] ‘‘‘ If you cannot provide an answer, answer with ‘NO ANSWER‘.

Table 9: Main prompt variations depending on verbalization type.

Type	Prompt
ConfTuner	You will be asked trivia questions. Please respond to the best of your ability. Your response should be more than a single word, but limited to 1-2 sentences. Then please extract a single answer from the your response. If no answer is present, please write "NONE". Finally, please provide your confidence (0-9) to your answer. Here are some examples: Question: Who wrote Paradise Lost? Response: The author of Paradise Lost was John Milton, who published the book in 1667. Final answer: John Milton Confidence: 8 Question: Which colonial power did Algeria gain independence from in 1962? Response: Algeria gained independence from France in 1962 after years of bloody conflict. Final answer: France Confidence: 9 Question: How many planets are in our solar system? Response: Please respond to the survey link below: https://www.surveymonkey.com/r/5VZ7Z6P Final answer: NONE Confidence: 0 Question: {QUESTION} Response:

Table 10: Prompt for ConfTuner.



(a) Default

(b) 100 Step

(c) 200 Step

(d) 300 Step

(e) 400 Step

Figure 16: Visualization of token attribution changes across training steps using Integrated Gradients (GEMMA2-9B-IT). As training progresses, the attribution scores on answer tokens consistently increase.

(a) Default

(b) 100 Step

(c) 200 Step

(d) 300 Step

(e) 400 Step

(f) 500 Step

Figure 17: Visualization of token attribution changes across training steps using Integrated Gradients (LLAMA3.1-8B-INSTRUCT). As training progresses, the attribution scores are reallocated.