Calibrating Long-form Generations from Large Language Models

Yukun Huang[1], **Yixin Liu**[2], **Raghuveer Thirukovalluru**[1],
Arman Cohan[2], **Bhuwan Dhingra**[1]

[1]Duke University, [2]Yale University
{yukun.huang, raghuveer.thirukovalluru}@duke.edu, bdhingra@cs.duke.edu
{yixin.liu, arman.cohan}@yale.edu

Abstract

To enhance Large Language Models' (LLMs) reliability, calibration is essential—the model's confidence scores should align with the likelihood of its responses being correct. However, traditional calibration methods typically rely on a binary true/false assessment of response correctness, unsuitable for long-form generations where an answer can be partially correct. Addressing this gap, we introduce a unified calibration framework, in which both the correctness of the LLMs' responses and their associated confidence levels are treated as distributions across a range of scores. We develop three metrics for assessing LLM calibration and propose confidence elicitation methods based on self-consistency and self-evaluation. Our experiments demonstrate that larger models don't necessarily guarantee better calibration, that various calibration metrics complement each other, and that self-consistency methods excel in factoid datasets. We also find that calibration can be enhanced through techniques such as fine-tuning, scaling the temperature. Finally, we illustrate one application of long-form calibration through selective answering in long-form responses, optimizing correctness within a constrained API budget.

1 Introduction

Confidence calibration in large language models (LLMs) aims to align the model's internal confidence with a probabilistic perspective of its answers' correctness (i.e. quality), enhancing reliability and interpretability for aiding human decision-making (Kadavath et al., 2022). People intuitively understand and utilize probabilities (Cosmides and Tooby, 1996), making this approach crucial for practical applications. Conventional calibration (Guo et al., 2017) treats answer correctness as binary (true or false) and seeks to align the model's confidence with the likelihood of model's answer being correct, typically stated as: "*I am x% confident that this answer is completely correct.*" However, the correctness of long-form generation is not always either true or false but can be partially correct (Figure 1).

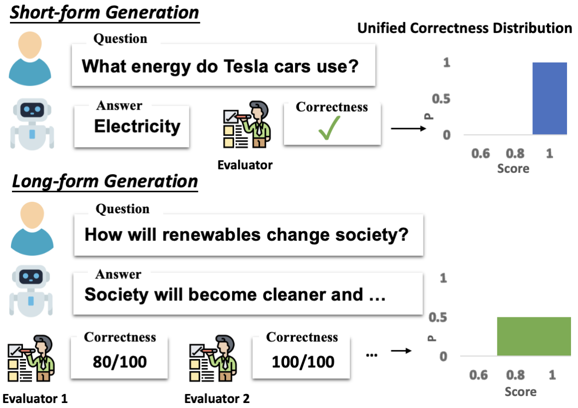


Figure 1: A comparison between short-form generation and long-form generation. The correctness of the short-form answer can either be true (1) or false (0), while the correctness of the long-form answer is typically a score between 0 and 1. Both of these scores may vary across evaluators due to subjectivity, hence we conceptualize them as a distribution over $[0, 1]$.

Therefore, a single confidence score for long-form outputs is ambiguous: it can either imply "*I am x% confident that the answer is 100% correct*" or "*I am 100% confident that the answer is x% correct.*" The former fails to capture the graded notion of long-form answer correctness, while the latter focuses on self-evaluation of correctness, rather than calibration as it overlooks the confidence at specific correctness levels.

Addressing this challenge, we propose to conceptualize the model's confidence as *distribution* across scores between $[0, 1]$ to capture the nuanced understanding of the model of each correctness level of the long-form answer, corresponding to the statement *"I am x% confident that this answer is y% correct"*. Moreover, we also view the correct-

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 13441–13460
November 12-16, 2024 ©2024 Association for Computational Linguistics

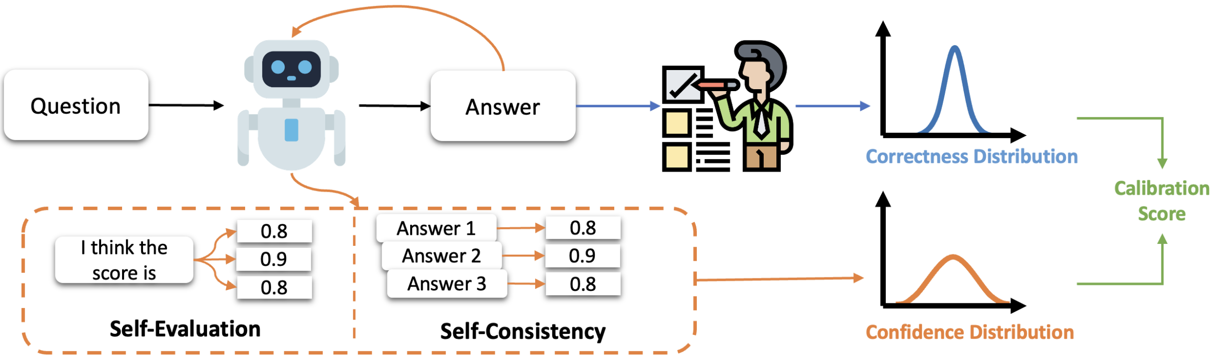


Figure 2: Overview of our calibration framework. We prompt an LLM to produce an answer to a specific question, assess the answer's correctness distribution using an evaluator (task-specific metric/GPT-4 metric/human metric), and determine the model's confidence distribution through self-evaluation or self-consistency approaches. Finally, we calculate the calibration score by comparing the correctness and confidence distributions against our predefined metrics.

ness of an LLM's response as a *distribution* across scores between $[0, 1]$ to capture graded and subjective assessments of long-form generations quality. This subjectivity arises from the multifaceted nature of evaluating long-form outputs, where factors like factuality, coherence, clarity, and comprehensiveness each play a role, potentially introducing variability in judgment (Bakker et al., 2022). We can then measure both the classical notion of calibration error, averaged across different correctness levels, as well as new notions of alignment between the correctness and confidence distributions and their utility in selective prediction (§ 3.4). Figure 2 shows an overview of our framework, which consists of three modular components: estimating the target correctness distributions, eliciting confidence distributions from LLMs, and measuring calibration between these distributions.

Our unified framework offers three key advantages. 1. Generalizability: Our framework applies to both long-form and short-form generation tasks by representing correctness and confidence as distributions, regardless of whether the correctness of task is binary, continuous, subjective, or objective. 2. Flexibility: The framework is evaluation-metric agnostic, allowing the integration of any metric or confidence elicitation method, and can adapt as evaluation methods evolve. 3. Interpretability: It provides a nuanced view of uncertainty, enabling decision-makers to assess confidence across multiple correctness levels, fostering greater transparency and trust in the model's outputs.

We leverage our framework to measure calibration for several LLMs on multiple datasets across three long-form QA—ASQA (Stelmakh et al., 2022), ELI5 (Fan et al., 2019), QAMPARI (Amouyal et al., 2022)—and one summarization task, CNNDM (Nallapati et al., 2016). Our results show that our methods excel over baselines by leveraging the model's nuanced confidence distribution, stronger LLMs like GPT-3.5 don't necessarily guarantee better calibration, that various calibration metrics complement each other, and LLMs exhibit better calibration on factoid datasets than more open-ended datasets. Furthermore, our analysis highlights that fine-tuning and temperature scaling could enhance calibration. Finally, we illustrate a practical application of long-form calibration: employing a cascading strategy (Chen et al., 2023a) for selective answering to optimize the cost-effectiveness of long-form text generation. In this approach, an open-source model initially handles queries and, based on its confidence levels—assessed using our system—a more advanced API model is engaged as needed. This method ensures cost efficiency while maintaining high-performance levels.

In summary, our contributions are:

- A universal calibration framework for text generation tasks, enhancing LLM evaluation for critical applications.
- Innovative methods for confidence elicitation and calibration measurement, applied to a variety of LLMs.
- Evidence that calibration can be improved by model fine-tuning and temperature scaling.
- A cost-effective model usage strategy, illustrating the practicality of long-form calibration in optimizing LLM deployment.

2 Related Work

Measuring Calibration Calibration (Guo et al., 2017, Minderer et al., 2021)has been widely studied in language models, whose probabilities derived from logits are generally found to not be calibrated (Jiang et al., 2020, Kadavath et al., 2022, Chen et al., 2023d). Standard metrics to measure the calibration include Expected Calibration Error (ECE) for confidence-accuracy disparity (Naeini et al., 2015), Brier Score for mean squared prediction-outcome differences, and AUROC for assessing confidence-based correct answer identification (Boyd et al., 2013, Kuhn et al., 2023). Selective Accuracy@Coverage measures accuracy within the model's most confident predictions (Liang et al., 2023, Cole et al., 2023). However, these metrics, rooted in a binary notion of correctness, fall short for long-form tasks where correctness spans a range, suggesting a distribution-based approach is more apt.

Improving Calibration Traditional calibration methods focus on post-processing logits (Guo et al., 2017), but with LLMs generating unbounded text, logits could fall short. Thus, extracting better confidence scores (i.e., confidence elicitation) has become crucial for improving calibration. These include: *verbalization*, which directly asks the model to output its confidence (Lin et al., 2022), *consistency*, which uses the uniformity of multiple responses to gauge confidence (Kadavath et al., 2022, Kuhn et al., 2023, Cole et al., 2023, Chen et al., 2023c, Tian et al., 2023a, Lin et al., 2023), and the *hybrid* of both (Xiong et al., 2023, Tian et al., 2023b, Chen and Mueller, 2023). However, these methods often presume binary answer correctness, offering a singular confidence score that fails to capture the nuanced correctness required for long-form tasks. Recent work (Zhang et al., 2024) on long-form generation addresses continuous correctness scores but focuses on aligning an uncertainty score with correctness, rather than on improving calibration.

3 Long-form Generations Calibration

This section formalizes the long-form generation calibration problem (§ 3.1), and introduces three core components in our calibration framework (Figure 2): the correctness distributions of the answers (§ 3.2), the confidence distributions of the LLM on its answers (§ 3.3), and the calibration metrics to measure how well these two align (§ 3.4).

3.1 Formulation

Given a dataset $\mathcal{D}$, the model's answer for each question Q_i in the dataset is answer A_i (where i indexes the questions in the dataset). To measure how calibrated the model is, we need three steps. First, we apply an evaluator to get the target correctness distribution P_{T_i} where T_i is the random variable that denotes the correctness score in answer A_i.

$$P_{T_i}(x) = \text{Pr}(A_i \text{ is } s \text{ correct}) \tag{1}$$

for $s \in \mathcal{S}$, where $\mathcal{S}$ is the space of correctness levels (e.g., normalized from ordinal scores ranging from 0 to 5). It should be noted that s could theoretically be a continuous value in the range $[0\%, 100\%]$. However, since humans tend to make more accurate judgments using discrete ratings due to "rounding bias" (Honda et al., 2022), we approximate continuous correctness with ordinal scores in practice. Second, We use a confidence elicitation method to derive the confidence distribution P_{C_i} from LLM $\mathcal{M}$, where C_i represents the model's confidence in its answer A_i. We ensure these confidence scores are normalized to form a valid distribution, matching the domain of the correctness distribution. For $s \in \mathcal{S}$,

$$P_{C_i}(x) = \mathcal{M}\text{'s confidence that } A \text{ is } s \text{ correct} \tag{2}$$

Finally, we design metrics to measure the alignment between P_{C_i} and P_{T_i} across the dataset.

3.2 Correctness Distribution Estimation

To establish correctness distributions as alignment targets, we need to adopt long-form evaluation metrics that integrate aspects such as relevance, coherence, factuality, and helpfulness. Traditional metrics like BLEU and ROUGE fail to capture semantic meaning (Liu et al., 2023a), while factuality-based metrics like FactScore (Min et al., 2023) may neglect question relevance. GPT-4 metrics have gained popularity (Li et al., 2024) due to their adaptability and comparative accuracy (Jain et al., 2023, Liu et al., 2023a). These metrics allow for the integration of various user-prioritized aspects by adjusting evaluation rubrics, providing a balanced approach to both referenced and divergent answers. However, they also have limitations like a bias toward longer outputs (Zheng et al., 2023).

Our framework is evaluator-agnostic, allowing us to use any correctness distribution metric. To identify a practical and actionable metric that best guides human decision-making, we conduct human evaluations to determine alignment with human judgment. According to the results in Appendix C, we determined that the GPT-4 metric, with a higher correlation to human judgments compared to the task-specific metric, is more effective for three datasets (e.g., 76.2 v.s 47.8 in ASQA), while the task-specific metric is preferred for another dataset. Our framework's modular design allows replacements of evaluation metrics based on user needs or new developments.

3.3 Confidence Distribution Elicitation

There are two common strategies to develop confidence in model responses: explicitly asking the model to verbalize its confidence or implicitly estimating it through self-consistency. However, the single confidence score provided by prior studies is ambiguous and non-interpretable in long-form calibration. Therefore, we develop two methods tailored for long-form calibration accordingly.

Continuous Self Evaluation (*CSE*) We prompt the model to repeatedly perform self-evaluations, where the resulting scores (typically ordinal scores from 0 to 5) are normalized to a [0,1] range and interpreted as a confidence distribution. The self-evaluation template closely mirrors the template for correctness evaluation but omits the reference answer (see Appendix F.3 for details). Formally, given an LLM $\mathcal{M}$, N self-evaluations:

$$P_{C_i}(s) = \frac{1}{N}\sum_{j=1}^{N} 1(\mathcal{M}(A_i)_j = s) \qquad (3)$$

for score $s \in \mathcal{S}$, where $\mathcal{S}$ is the space of correctness levels. Such a sampling method provides a more authentic reflection of the model's internal distribution than logits (Cole et al., 2023). By asking the model to assess an answer multiple times, we capture a range of scores that better represent the model's confidence, which enhances the reliability of the confidence estimation.

Pairwise Self Consistency (*PSC*) Another key indicator of model confidence is the consistency among multiple responses a model provides for a given question. Given a primary answer A_i, other N answers $A_i^1 ... A_i^N$ sampled from LLM $\mathcal{M}$, a metric for measuring the similarity between two answers $Sim(\cdot,\cdot)$, and a score $s \in \mathcal{S}$:

$$P_{C_i}(s) = \frac{1}{N}\sum_{j=1}^{N} 1(Sim(A_i, A_i^j) = s) \qquad (4)$$

Assessing similarity in long-form answers is more complex than with short responses (Kadavath et al., 2022). To address this, we propose four methods for evaluating similarity in long-form content: 1. **Naive**, assessing overall response similarity with an LLM; 2. **Split**, analyzing sentence-level similarity; 3. **Claim**, evaluating claim matching; 4. **Named Entity Recognition** (NER), focusing on named entity overlap. These approaches range from broad to detailed analysis, chosen based on task requirements and the desired analysis depth. See Appendix B for more details.

3.4 Calibration Metrics

We introduce three key metrics to assess model calibration from various angles. *Expected correctness error with multi-class* (ECE-M) measures the alignment between the model's stated confidence in reaching a particular level of correctness and the actual likelihood that the model performs at that specified level, across the spectrum from 0 to 1. *Correlation* evaluates the alignment between expected confidence and correctness *across the dataset*, indicating the model's proficiency in ranking answers. *Selective F1* measures the utility of confidence scores in identifying the good answers and abstaining from the rest.

ECE-M The classical notion of calibration relies on an (answer-correctness) pair of random variables $(A, Y) \in \mathcal{A} \times \{0,1\}$, where $\mathcal{A}$ is the answer space. An LLM $\mathcal{M}$ with its confidence elicitation method $h_{\mathcal{M}}$: $\mathcal{A} \to [0,1]$ is said to be well-calibrated if $\Pr(Y = 1|h_{\mathcal{M}}(A) = q) = q$ for $q \in [0,1]$. To measure if this holds, traditional ECE(h) (Gupta and Ramdas, 2021) is: $\mathbb{E}_A\left[|\Pr(Y = 1|h_{\mathcal{M}}(A)) - h_{\mathcal{M}}(A)|\right]$

In long-form calibration where the answer correctness is a continuum $Y \in [0,1]$, the probabilistic confidence predictor $h_{\mathcal{M}}$ should predict confidence about each level of correctness and therefore denoted as $h_{\mathcal{M}}$: $\mathcal{A} \times [0,1] \to [0,1]$. The notion of long-form calibration is $\Pr(Y = s|h_{\mathcal{M}}(A, s) = q_s) = q_s$ for every $s \in [0,1]$ and $q_s \in [0,1]$ We define ECE-M as the aggregation of ECE scores for all correctness levels. In practice, we use discrete levels $s \in \mathcal{S}$ (e.g., ratings from 0-5) for the correctness scores. Hence, we calculate an ECE(s, h) conditioned on each s,

$$\mathbb{E}_X\left[|\Pr(Y = s|h_{\mathcal{M}}(A, s)) - h_{\mathcal{M}}(A, s)|\right] \qquad (5)$$

Then the final ECE-M score is weighted by the frequency of each class:

$$\text{ECE-M}(h) := \sum_{s \in \mathcal{S}} \Pr(Y = s)\text{ECE}(s, h) \quad (6)$$

Correlation While ECE-M focuses on measuring calibration at each correctness level independently, it doesn't account for the distance between these levels. To address this, we also measure the correlation between the expected values of the confidence and correctness distributions for a more comprehensive assessment. For each answer A_i within the dataset, we can calculate the expected correctness score of the model $E|C_i| = \sum_s P_{C_i}(s) \times s$ and the expected correctness score of the target $E|T_i| = \sum_s P_{T_i}(s) \times s$. In the whole dataset, we can get a list of expected correctness scores $\mathbf{E_C}$ from model confidence and a list of target expected correctness scores $\mathbf{E_T}$. Then we can measure the correlation between them.

$$\rho(\mathcal{D}) = Corr(\mathbf{E_C}, \mathbf{E_T}) \quad (7)$$

where $Corr(\cdot)$ represents the correlation function. This correlation provides a clear indicator of how well our model's confidence aligns with its actual correctness across the entire dataset.

Selective F1 In selective answering (e.g., Kamath et al., 2020), models only respond when confident about their accuracy to improve reliability. Traditional metrics for selective answering include accuracy@coverage and coverage@accuracy (Tian et al., 2023b), which measures a model's precision and recall in selecting completely correct answers. Similarly, in long-form calibration, it is crucial to assess the model's selection of answers that are at least "s% correct" using both precision and recall.

Therefore, we propose the selective F1 metric ($F1_{\tau_s}$) to quantify the model's aptitude in filtering out answers that meet or exceed a predefined correctness threshold τ_s. Our approach utilizes a dual-threshold system, consisting of a confidence threshold (τ_c) and the correctness threshold (τ_s), allowing the model to answer questions only if its confidence in the answer's expected correctness score exceeding τ_s surpasses τ_c. Formally, let $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ denote the total set of model's answers in the dataset and $\mathcal{A}^* = \{A_i \in \mathcal{A} \mid \sum_{s \geq \tau_s} P_{C_i}(s) \geq \tau_c\}$ denote the set of selected answers. Let the indicator function $I_{\tau_s}(A_i)$ indicate if the expected correctness score $E[T_i]$ of A_i exceeds τ_s:

$$I_{\tau_s}(A_i) = \begin{cases} 1 & \text{if } E[T_i] \geq \tau_s \\ 0 & \text{Otherwise} \end{cases} \quad (8)$$

The selective precision P_{τ_s} on the dataset $\mathcal{D}$ is the proportion of selected answers that surpass the correctness threshold τ_s relative to the total number of selected answers:

$$P_{\tau_s}(\mathcal{D}) = \frac{\sum_{A_i \in \mathcal{A}^*} I_{\tau_s}(A_i)}{|\mathcal{A}^*|} \quad (9)$$

The selective recall R_{τ_s} compares the number of selected answers meeting this criterion against the total number of correct answers in the dataset that exceed the threshold τ_s:

$$R_{\tau_s}(\mathcal{D}) = \frac{\sum_{A_i \in \mathcal{A}^*} I_{\tau_s}(A_i)}{\sum_{A_i \in \mathcal{A}} I_{\tau_s}(A_i)} \quad (10)$$

The selective F1 combines recall and precision:

$$F1_{\tau_s}(\mathcal{D}) = 2\frac{P_{\tau_s}(\mathcal{D})R_{\tau_s}(\mathcal{D})}{P_{\tau_s}(\mathcal{D}) + R_{\tau_s}(\mathcal{D})} \quad (11)$$

In our experiments, we select τ_s as the nearest correctness level greater than the best LM's average correctness score. For τ_c, we choose the value that yields the highest selective F1 score on the development split.

4 Experiments and Results

4.1 Setup

Models and Data We measure different sized LLMs' calibration, including Llama-2-13b-chat, Llama-2-70b-chat (Touvron et al., 2023), Vicuna-13b (Zheng et al., 2023), Llama-3-8b-Instruct, GPT-3.5-turbo, across three long-form QA tasks: ASQA (Stelmakh et al., 2022), ELI5 (Fan et al., 2019), QAMPARI (Amouyal et al., 2022), and one summarization task: CNNDM (Nallapati et al., 2016). Details of datasets can be found in Appendix A.

Correctness Evaluation We apply GPT-4 to evaluate target correctness distributions for ASQA, ELI5, and CNNDM. In QAMPARI where the answer is a list of entities, we evaluate using the F1-5 metric, calculating the F1 score by the exact match with the gold answer and defining 100% recall for predictions with at least 5 correct answers.

4.2 Confidence Elicitation Methods

In addition to our methods *CSE* and *PSC* (see Appendix D.3 for similarity measurement choosing), we established baselines for self-evaluation, self-consistency, and logits-based approaches. This is

	ASQA			QAMPARI			ELI5			CNNDM		
Method	ECE-M	Corr	$F1_{0.8}$	ECE-M	Corr	$F1_{0.4}$	ECE-M	Corr	$F1_{0.8}$	ECE-M	Corr	$F1_{0.8}$
SL*	28.2	0.7	0.0	27.4	10.9	8.0	29.6	-11.9	0.0	77.2	-7.5	0.0
BSE*	32.8	14.2	57.6	**25.2**	16.8	33.2	30.3	11.7	46.7	78.5	11.2	90.6
CSE	29.0↓	16.3↑	58.5 ↑	42.8↑	21.9↑	33.4↑	31.2↑	**26.9**↑	**48.2**↑	**15.2**↓	**19.2**↑	**92.0**↑
ASC*	35.9	27.1	5.2	46.0	38.6	38.5	38.4	16.7	7.9	63.2	8.8	44.2
PSC$_{F1}$	28.8	27.1	33.5	38.4	38.6	42.7	27.1	16.7	20.7	57.1	8.8	79.5
PSC	**18.3**↓	**46.8**↑	**61.6** ↑	26.2↓	**39.1**↑	**44.0**↑	**24.9**↓	24.9↑	46.2↑	64.5↑	15.5↑	90.0↑

Table 1: Calibration Performance Comparison Among Different Confidence Elicitation Methods Across Four Tasks (in %): "ECE-M" for expected correctness error with multi-class, "Corr" for Correlation, "$F1_{\tau_s}$" for Selective F1 Score at threshold τ_s. Results represent averages from five models. Methods with * served as baselines. For "Corr", "$F1_{\tau_s}$", and "Score", ↑ means better than corresponding baseline while ↓ is worse. For "ECE-M", ↓ is better while ↑ is worse. The best score among all confidence elicitation methods is **bolded**. Key insights: 1) Self-Consistency (PSC) outperforms Self-Evaluation on factoid datasets; 2) Our methods PSC and CSE surpass baselines; 3) Different metrics offer complementary insights

because prior studies lack directly applicable baselines, primarily due to the non-interpretable nature of single confidence scores.

Sentence Likelihood (*SL*): Based on prior studies using logits to gauge model confidence, we adopt sentence likelihood as a baseline measure, which typically results in a confidence distribution focused at the lowest score in long-form answers.

Binary Self-Evaluation (*BSE*): following previous work (Kadavath et al., 2022) that asks model to self-evaluate if its answer is true several times, using the frequency of true as model's confidence score towards the answer being true. Then we adapt such a single score as a distribution focus solely on the values 0 and 1.

Average Self-Consistency (*ASC*): Following prior work (Xiong et al., 2023) using the average consistency between these candidate responses and the original answer then serves as a single measure of confidence score, we adopt the simple f1-token score to measure the consistency to adapt it to long-form generations. Then we treat the single score as a point mass distribution.

Pairwise Self-Consistency F1 (*PSC-F1*): Still using F1 to measure the consistency like *ASC*, but we directly treat the pairwise consistency scores as a distribution without aggregating, thereby keeping model intrinsic understanding about different correctness levels.

4.3 Main Results

In Table 1, we evaluate the calibration performances of various confidence elicitation methods by averaging the scores across all models. Table 1 shows that our methods, *CSE* and *PSC*, generally outperform their respective baseline categories and also surpass the logits-based method *SL*. Key findings from the results include:

Self-Consistency Outperforms Self-Evaluation on Factoid Datasets Self-consistency methods typically outperform self-evaluation on factoid datasets like ASQA and QAMPARI. However, their effectiveness diminishes in more subjective tasks such as ELI5 or CNNDM. We hypothesize that this is because self-consistency is more readily quantifiable in factoid datasets, where the agreement between answers can be assessed based on factual consistency, thus providing clearer criteria. Conversely, in open-ended datasets, the consistency between answers is more ambiguous, making it more difficult to measure.

Nuanced Self-Evaluation Enhances Calibration *CSE* generally outperforms *BSE* by providing detailed confidence estimates at each correctness level. However, the overall improvement remains constrained by the intrinsic limitations of LMs in self-evaluating their correctness, which sometimes hampers accurate estimations.

Pair-wise Similarities Distribution and Task-tailored Similarity Measurement Help Calibration Both *ASC* and *PSC-F1* measure similarity with token-level F1. However, *PSC-F1* treats these scores as a distribution rather than averaging them, leading to better ECE-M and selective F1. *PSC* further enhances calibration by adopting a task-specific, detailed measurement of similarity, outperforming FSC in all four tasks.

	ASQA				QAMPARI			
Model	ECE-M	Corr	$F1_{0.8}$	Score	ECE-M	Corr	$F1_{0.4}$	Score
Llama2-13b	15.9	48.0	47.5	51.3	30.5	46.0	42.3	13.3
Llama2-70b	14.7	44.3	61.9	59.4	29.3	17.1	37.0	14.6
Vicuna-13b	20.1	**58.2**	56.8	50.8	**14.4**	**49.5**	42.6	11.4
Llama-3-8b	**14.3**	53.2	65.1	54.9	33.0	38.3	42.0	14.1
GPT-3.5-turbo	26.7	30.5	**76.7**	**72.6**	23.7	44.4	**56.2**	**24.0**

	ELI5				CNNDM			
Model	ECE-M	Corr	$F1_{0.8}$	Score	ECE-M	Corr	$F1_{0.8}$	Score
Llama2-13b	36.0	21.8	40.1	53.8	12.6	19.5	92.0	77.0
Llama2-70b	32.8	18.8	54.4	61.7	13.8	6.4	93.6	77.6
Vicuna-13b	**19.4**	31.3	34.9	53.0	32.7	3.9	64.2	**78.2**
Llama-3-8b	34.9	**36.7**	48.4	57.1	9.3	**49.0**	86.9	77.8
GPT-3.5-turbo	32.7	26.2	**63.4**	**63.0**	**7.8**	17.0	**94.7**	**78.2**

Table 2: Comparison of Calibration Performance Across Models for Four Tasks (in %): We identify the optimal confidence elicitation method for each task and compare the performance of various models using this method. "Score" means the model's average correctness score on that task. A key observation is that more powerful LMs do not necessarily exhibit better calibration, although they tend to perform better in selective answering.

Calibration metrics complement each other A simplistic approach like *SL*, which allocates all the probability mass to the point of score 0, can misleadingly show decent ECE-M (28.2%) in specific cases like ASQA. However, its negative correlation (0.7%) and zero $\mathrm{F1}_{0.8}$ underscore an ineffective confidence distribution. Similarly, BSE in CNNDM may achieve a high $\mathrm{F1}_{0.8}$ (90.6%) by overestimating answer correctness, but this does not truly reflect response quality (correlation: 11.2%) or provide well-calibrated probabilities across correctness levels, resulting in a bad ECE-M (ECE-M: 78.5%). Hence, a comprehensive evaluation using multiple metrics is essential for a balanced assessment of model calibration.

Larger models are not necessarily better calibrated. In Table 2, we focus on the calibration performance of individual models when paired with the best-performing confidence elicitation method for each task. Table 2 shows that despite poor performance on ASQA and QAMPARI, Vicuna-13b has the highest correlation across these datasets. It might be because reinforcement learning for other models causes miscalibration by encouraging overfitting to rewarded behaviors (Kadavath et al., 2022). Scaling the temperature could enhance the calibration of LLMs fine-tuned using RL (see § 4.4). Additionally, Llama-2-13b demonstrates a higher correlation than its larger counterpart, Llama-2-70b. However, GPT-3.5-turbo, the strongest model, consistently scores the highest in selective F1 across all datasets. This performance can be attributed to the model's ability to generate a larger volume of high-quality answers, increasing the probability of selecting superior responses even if it is not particularly well-calibrated. Consequently, the Selective F1 metric blends performance and calibration, and tends to favor more capable models due to their higher output of quality answers.

4.4 Improving Calibration

We delve into different strategies to enhance calibration: fine-tuning, scaling the temperature, adding source documents (Appendix E.2), and hybrid confidence elicitation (Appendix E.3).

Fine-tuning Our study explores three fine-tuning strategies to improve model calibration on the ASQA dataset: fine-tuning the model for self-evaluation (using questions and model answers to produce scores and explanations), fine-tuning the model for generation (generating answers from questions), and a hybrid of both. GPT-4 synthesizes self-evaluation data by assessing different models' answers to questions from the ASQA training set, while the generation data originates from the ASQA training set itself. We apply LoRA (Hu et al., 2021) fine-tuning to the Llama2-13b model. See Appendix E.1 for experiment details. As Table 3 reveals, solely training on self-evaluation ('Evaluation') did not yield consistent improvements in calibration, possibly due to the complexity of this task and the limitation of LORA. Nonetheless, fine-tuning the model improves the self-consistency method, especially when the generation data is included during training ('Eval + Gen' and 'Generation'). The model becomes more confident in terms of self-consistency after fine-tuning.

Data	Corr	ECE-M	$\mathbf{F1}_{0.8}$	Score
Self-Evaluation (CSE)				
None	18.1	30.3	50.8	51.3
Evaluation	13.6 ↓	32.4 ↑	52.3 ↑	49.2 ↓
Generation	20.0 ↑	26.2 ↓	53.4 ↑	52.1 ↑
Eval + Gen	23.9 ↑	20.2 ↓	46.6 ↓	50.1 ↓
Self-Consistency (PSC)				
None	48.0	15.9	47.5	51.3
Evaluation	46.9 ↓	13.6 ↓	56.1 ↑	49.2 ↓
Generation	58.9 ↑	14.5 ↓	59.5 ↑	52.1 ↑
Eval + Gen	54.5 ↑	12.2 ↓	50.7 ↑	50.1 ↓

Table 3: Comparison among raw and fine-tuned Llama-2-13b on ASQA. "None" for the untrained model, "Evaluation" for the model fine-tuned with the self-evaluation dataset, "Generation" for the model fine-tuned with the ASQA generation data, and "Eval+Gen" for the model fine-tuned with the hybrid dataset combined by self-evaluation dataset and generation data.

Temperature We adjust the generation temperature for Llama2-13b from 0.2 to 1 to examine its impact on calibration. The result in Figure 3 reveals consistent improvements in all calibration metrics. Notably, the model's performance initially improves and then deteriorates. This observation implies that modulating the generation temperature can enhance the calibration of the model.

4.5 Application

We showcase an application of long-form calibration in Figure 4: a cost-effective cascading strategy using language models of varying capabilities to efficiently handle queries within an API budget constraint. Initially, an open-source model (Llama-2-13b in our experiment) address questions where it believes the answer has a probability higher than τ_c that the answer's correctness score is above τ_s. Complex queries, flagged by lower model confidence, are escalated to a more advanced API LM (GPT-4). Adjusting τ_c between 0 and 1 controls how many queries reach GPT-4, balancing answer quality with API budget constraints. We benchmark using the open-source LM for a zero API budget and the commercial LM for full-budget scenarios. Our experiments utilize the ASQA and QAMPARI datasets to evaluate four distinct confidence elicitation strategies: PSC, ASC, CSE, and BSE. Additionally, we incorporate a baseline strategy where, under a constrained number of API requests, a random selection of queries is processed by Llama-2-13b, with the remaining handled by GPT-4. For each API budget scenario, questions are randomly assigned to Llama-2-13b using 10 different random seeds, and we calculate the mean and standard deviation of the results. We focus on the success rate, which we define as the percentage of answers that meet or exceed a user-specified score threshold. This metric is reported both for the overall dataset and for the subset of queries selected and handled by Llama-2-13b, illustrating both the general effectiveness of our cascading model and the selective answering capabilities of the individual model. As shown in Figure 5, PSC generally outperforms the other methods, with CSE and BSE yielding comparable results that follow. ASC, in contrast, performs the poorest, comparable to the random selection strategy. These results highlight the pivotal role of advanced calibration techniques, confidence elicitation methods in our case, in boosting the practical utility and cost-efficiency of LLMs when API usage is limited.

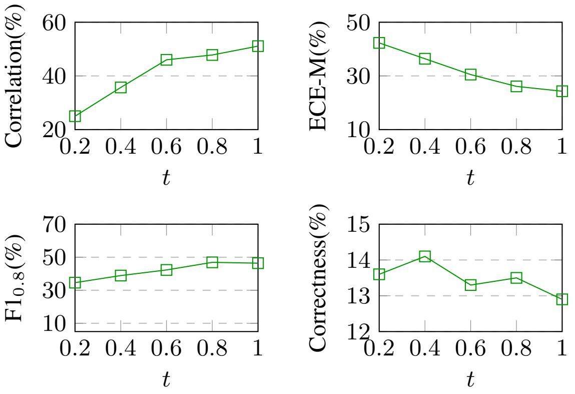


Figure 3: Calibration varies with temperature scaling.

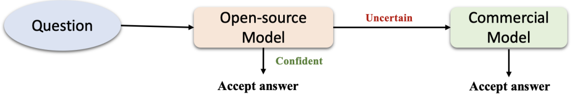
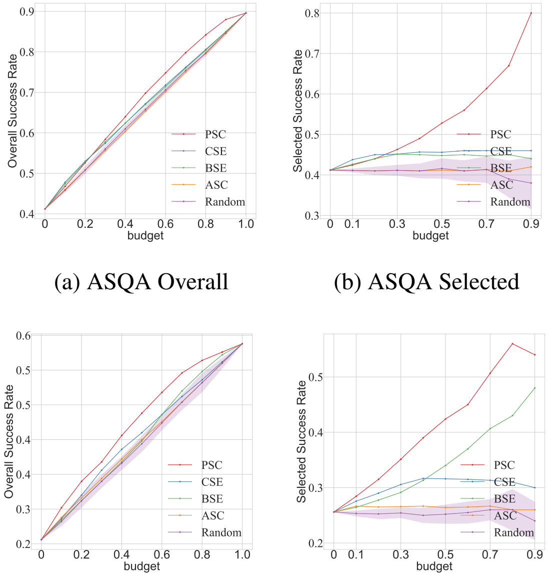


Figure 4: The illustration of LLM Cascade.



(a) ASQA Overall

(b) ASQA Selected

(c) QAMPARI Overall

(d) QAMPARI Selected

Figure 5: Variation in Success Rate by API Budget Allocation on the ASQA and QAMPARI Datasets for All Queries and Those Selected by Llama-2.

5 Conclusion

Our study presents a novel calibration system for evaluating LLMs in long-form generation. Our results challenge the assumption that larger LLMs are always calibrated better and show calibration variability across datasets. Additionally, we propose methods to improve LLM calibration and show an application that optimizes performance under API budget constraints. The system we present is crucial for further improving the liability of LLMs.

6 Limitation

Our study faces three primary limitations. First, we rely on GPT-4 to estimate the target correctness distribution. But as tasks become more subjective, consensus on humans' evaluations may decrease. This wider target correctness distribution becomes challenging for both GPT-4 and human annotators to accurately capture. This limitation is inherent to natural language generation (NLG) evaluation and lies beyond the purview of our project. Our framework operates under the premise that a target correctness distribution exists and concentrates on calibration which aligns the model's confidence with this assumed target. Second, our experiments focus on long-form QA and do not extend to specialized domains such as law, medicine, or education, where the calibration of LLMs could have significant real-world implications. Lastly, our self-consistency method is computationally intensive, posing a challenge for practical applications. There is a need for more efficient approaches in real-world settings.

Replicability:
Codes: `https://github.com/kkkevinkkkkk/calibration`

References

Samuel Joseph Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. 2022. Qampari: An open-domain question answering benchmark for questions with many answers from multiple paragraphs.

Michiel A. Bakker, Martin Chadwick, Hannah Sheahan, Michael Henry Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nathan McAleese, Amelia Glaese, John Aslanides, Matthew M. Botvinick, and Christopher Summerfield. 2022. Fine-tuning language models to find agreement among humans with diverse preferences. *ArXiv*, abs/2211.15006.

Kendrick Boyd, Kevin H. Eng, and David Page. 2013. Area under the precision-recall curve: Point estimates and confidence intervals. In *ECML/PKDD*.

Jiuhai Chen and Jonas Mueller. 2023. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness.

Lingjiao Chen, Matei A. Zaharia, and James Y. Zou. 2023a. Frugalgpt: How to use large language models while reducing cost and improving performance. *ArXiv*, abs/2305.05176.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023b. Universal self-consistency for large language model generation. *ArXiv*, abs/2311.17311.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2023c. On the relation between sensitivity and accuracy in in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 155–167, Singapore. Association for Computational Linguistics.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2023d. A close look into the calibration of pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1343–1367, Toronto, Canada. Association for Computational Linguistics.

Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions.

In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, Singapore. Association for Computational Linguistics.

Leda Cosmides and John Tooby. 1996. Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty. *Cognition*, 58:1–73.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *ArXiv*, abs/2209.12356.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*.

Chirag Gupta and Aaditya Ramdas. 2021. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *ArXiv*, abs/2006.03654.

Hidehito Honda, Rina Kagawa, and Masaru Shirasuna. 2022. On the round number bias and wisdom of crowds in different response formats for numerical estimation. *Scientific Reports*, 12.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8487–8495, Toronto, Canada. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Saurav Kadavath, Tom Conerly, Amanda Askell, T. J. Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *ArXiv*, abs/2207.05221.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Joonghoon Kim, Saeran Park, Kiyoon Jeong, Sangmin Lee, Seung Hun Han, Jiyoon Lee, and Pilsung Kang. 2023. Which is better? exploring prompting strategy for llm-based metrics. *ArXiv*, abs/2311.03754.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *ArXiv*, abs/2302.09664.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024. Leveraging large language models for nlg evaluation: A survey. *ArXiv*, abs/2401.07103.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher R'e, Diana Acosta-Navas, Drew A. Hudson, E. Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan S. Kim, Neel Guha, Niladri S. Chatterji, O. Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas F. Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525:140 – 146.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *Trans. Mach. Learn. Res.*, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Ann Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. *ArXiv*, abs/2106.07998.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2023a. Fine-tuning language models for factuality. *ArXiv*, abs/2311.08401.

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

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

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *ArXiv*, abs/2306.13063.

Caiqi Zhang, Fangyu Liu, Marco Basaldella, and Nigel Collier. 2024. Luq: Long-text uncertainty quantification for llms. *ArXiv*, abs/2403.20279.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori Hashimoto. 2023. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong

Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685.

A Dataset

ASQA (Answer Summaries for Questions which are Ambiguous) (Stelmakh et al., 2022) is a specialized long-form factoid dataset, designed to address ambiguous factoid questions that yield different correct answers based on their interpretations. This dataset challenges models to synthesize factual information from multiple sources, creating coherent long-form summaries that effectively resolve the inherent ambiguities in these questions.

ELI5 (Fan et al., 2019) is a comprehensive open-ended long-form dataset, encompassing over 270,000 threads from the Reddit forum "Explain Like I'm Five." This unique platform features community-generated responses to a wide array of questions, all tailored to be easily understandable by a five-year-old audience. The majority of queries in ELI5 are centered around 'how,' 'why,' and 'what' questions, which necessitate comprehensive, detailed responses supported by evidence from multiple passages.

QAMPARI (Amouyal et al., 2022) is a factoid dataset where answers are presented as lists of entities dispersed across multiple paragraphs. Its construction involves an automated process that utilizes Wikipedia knowledge graphs and tables. Questions are manually paraphrased, and answers are thoroughly verified for accuracy. Notably, each question in QAMPARI is associated with an average of 13 answers, demonstrating its breadth.

CNNDM (Nallapati et al., 2016) is a large-scale news summarization dataset containing news articles from CNN[1] and DailyMail[2]. The original CNNDM dataset consists of both source news articles and reference summaries. However, recent work (Liu et al., 2023b; Zhang et al., 2023) has found that the provided reference summaries are not of very good quality and zero-shot LLMs summaries are preferred by human annotators over the reference summaries.

In the **ACLE** (Automatic LLMs' Citation Evaluation) (Gao et al., 2023), a pioneering benchmark for assessing LLMs' citation capabilities, a subset of 1,000 examples is randomly selected from the development sets of ASQA, ELI5, and QAMPARI to form a test set for each task. For our specific analysis, we choose to utilize the first 500 examples from each of these datasets in ACLE as our test set, providing a focused and representative sample for each task. For CNNDM, we utilize 100 examples as our test set.

B Self-consistency

We propose four different self-consistency based methods tailored for long-form generation, each with a different strategy to measure the similarity between two long-form answer.

Naive The most basic approach utilizes an additional LLM (GPT-3.5-turbo in our experiments, which can be replaced by other models trained for this task) to determine if two responses are akin, assigning a corresponding similarity score. This method diverges from the relevant technique in contemporary research (Chen et al., 2023b), which primarily focuses on identifying the most consistent answer. Instead, our approach aims to secure specific consistency ratings that reflect the model's assurance in its primary answer, offering a general overview of the answers' similarity. The template for similarity measuring can be found in Appendix F.4.

Sentence Split For a more detailed similarity analysis between the two answers, we split the first answer into individual sentences. Another LLM (GPT-3.5-turbo or a similar NLI model) is then used to evaluate whether similar statements are present in the second answer. This method's limitation is that not all sentences carry equal informational weight. Some may be filler or less informative, potentially skewing the similarity assessment. The template can be found in Appendix F.4.

Claim To further refine the approach, we focus on sentences that make factual claims. This involves two steps: first, using a claim detector to identify factual claims within a sentence and then using an NLI model to determine if similar factual claims exist in the second answer. This method operates under the assumption that factual

[1] `https://www.cnn.com/`

[2] `https://www.dailymail.co.uk/`

claims are the most critical components of an answer, representing its core information. We leverage a DeBERTa-V2 (He et al., 2020) trained by ClaimBuster as our fact detector and GPT-3.5 as the NLI models.

Named Entity Recognition Advancing the granularity further, we compare named entities between two responses. We identify and compare entities present in both answers by utilizing a named entity recognition model. The degree of overlap in these entities serves as an indicator of answer similarity. This approach focuses more on concrete, identifiable elements within the answers. We use a Roberta-large (Liu et al., 2019) trained with SpanMarker framework [3], which can be replaced by other NER models.

C Evaluation Metric

C.1 GPT-4 metric

We ask GPT-4 to range an answer from 0 (worst) to 5 (best), which is then normalized to [0, 1]. See Appendix F.1 for details of the template. To mitigate scoring variability from criteria ambiguity and LLM uncertainty, we have evaluators repeatedly score each answer, forming a score distribution that better reflects its correctness. Specifically, given an LLM evaluator $\mathcal{E}$, N evaluations from it, and a score $x \in [0, 1]$, the correctness distribution is:

$$P_{T_i}(x) = \frac{1}{N}\sum_{j=1}^{N} 1(\mathcal{E}(A_i)_j = x) \qquad (12)$$

C.2 Human Evaluation

We utilize GPT-4 to assess answers across different tasks, including ASQA, ELI5, and CNNDM, using GPT-4 scores as a proxy for the target distribution. To demonstrate the better alignment of GPT-4 scores with human preferences over task-specific metrics, we focus our human evaluation efforts on the long-form tasks of ASQA and ELI5. This approach is supported by prior research indicating GPT-4's congruence with human judgments on summarization tasks (Liu et al., 2023a), thereby obviating the need for manual evaluation of CNNDM. Following Gao et al. (2023), in ASQA, we adopt the *EM-recall* automatic metric, which gauges the recall of correct short answers by verifying if the dataset's provided short answers are exact substrings of the generated content, following established methodologies. For ELI5, we utilize the most precise automatic metric to date, *claim recall*, employing the TRUE (Honovich et al., 2022) natural language inference model to ascertain if the generated output encompasses the sub-claims of the reference answer.

We present the task criteria to humans and ask them to provide a score for each answer based on the criteria. Participants are provided with a reference answer—not as an exclusive ground truth but as a guide—and are permitted to use search engines for additional context. We enlisted three annotators to evaluate 75 samples each for ASQA and ELI5, and calculate the average of them as a human score. Our analysis compares these human scores with those generated by task-specific metrics and GPT-4. The results, as detailed in our Table 4, underscore GPT-4's closer alignment with human judgments in both ASQA and ELI5. As shown in Figure 12, as the task becomes more open-ended like Eli5, the human agreements become lower than the factoid dataset ASQA. This further evidence the assumption that the correctness of a long-form answer should be a distribution.

	ASQA		**ELI5**	
Metric	**EM**	**GPT-4**	**Claim**	**GPT-4**
Corr ↑	47.8	76.2	42.9	71.5
MAE ↓	43.0	12.9	52.0	9.1

Table 4: Comparison of Human, Task-Specific metric, and GPT-4 Correctness Distributions in ASQA and ELI5 Tasks. Results are with %. 'Corr' denotes Pearson correlation (the higher the better), 'MAE' denotes mean absolute error (the lower the better). 'EM' denotes EM-recall, and 'Claim' denotes Claim-recall.

D Main Experiment

We assess the calibration of variously sized models (Llama-2-13b, Llama-2-70b, Vicuna-13b, GPT-3.5-turbo) across three long-form QA datasets (ASQA, ELI5, QAMPARI) and one summarization dataset (CNNDM). The process involves several steps: 1. Generation: Asking the model to generate answers for questions in the dataset 2. Correctness Assessment: We utilize GPT-4 to evaluate the correctness of the models' answers, except for QAMPARI, where we directly apply the F1-5 metric. 3. Confidence Distribution Derivation: After generating answers, models employ self-evaluation or self-consistency methods to derive their confidence

[3] https://github.com/tomaarsen/SpanMarkerNER

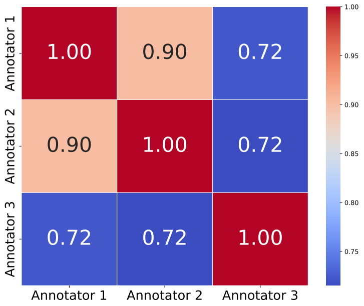


Figure 6: Correlation for ASQA

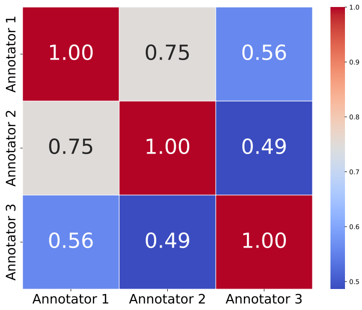


Figure 7: Cohen Kappa for ASQA

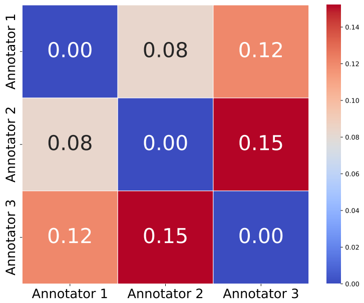


Figure 8: MAE for ASQA

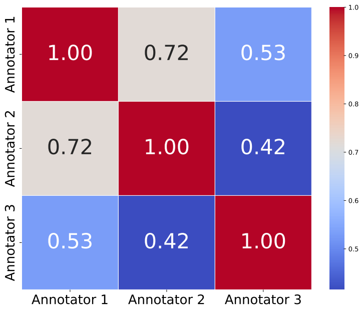


Figure 9: Correlation for ELI5

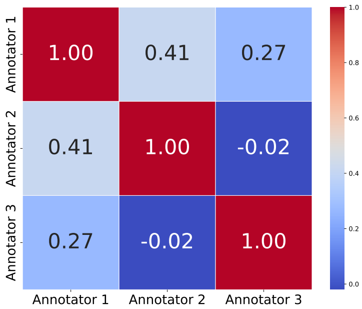


Figure 10: Cohen Kappa for ELI5

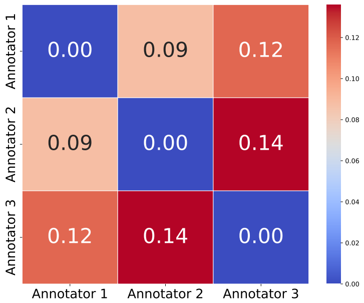


Figure 11: MAE for ELI5

Figure 12: Pairwise annotator agreements for ASQA and ELI5 respectively

distribution. 4. Calibration Measurement: .

D.1 Generation

In the answer generation phase, we employ a 3-shot in-context learning approach for each long-form QA dataset, providing three exemplars to guide the models. For the CNNDM dataset, we adopt a 0-shot strategy, aligning with prior studies (Goyal et al., 2022). The generation of answers utilizes a top-K sampling method, setting the generation temperature for all models at 0.6 and the top-K parameter at 10.

D.2 Correctness Evaluation

For ASQA, ELI5, CNNDM, we ask the GPT-4 to evaluate the correctness of examples. For ASQA and ELI5, We ask GPT-4 to evaluate the answer three times, producing a distribution capturing criteria ambiguity and model's inherent subjectivity. However, for CNNDM, we only ask GPT-4 to evaluate once to save computation given that most answers correctness concentrates around 0.8 with small variance. The GPT-4 evaluation template can be found in Appendix F.1 and criteria for different tasks can be found in Appendix F.2. For QAMPARI, which involves generating a list of entities as answers, we determine that F1 scores provide a more suitable measure of correctness than GPT-4 evaluations. To this end, we adopt the F1-5 metric (Gao et al., 2023), computing the F1 score based on direct matches with the correct answer list and assigning a recall of 100% for responses containing at least five accurate answers (recall-5).

D.3 Confidence Elicitation

After getting the answer from the model, we leverage the self-evaluation or self-consistency method to derive confidence distribution from the model.

Self-Evaluation We prompt the model to self-evaluate an answer 10 times, creating a confidence distribution based on its self-evaluation scores. Each self-evaluation includes task instructions, grading criteria, and evaluation examples (three for all long-form QA datasets and one for CNNDM due to length constraints). Additionally, it contains specific instructions for self-evaluation, incorporating both the question and the answer under evaluation. The detailed self-evaluation template and criteria are available in Appendix F.3 and Appendix F.2 respectively.

Self-consistency In the self-consistency approach, we generate an answer to the same question 10 times, designating the first response as the primary answer. We then calculate similarity scores between the primary answer and the remaining responses. These scores collectively create the model's confidence distribution for the primary answer. For various datasets, tailored strategies are employed to compare similarities between two answers.

For ASQA, we employ a self-consistency-claim approach. To assess the similarity between the two answers, we first identify factual claims in the first answer using a ClaimBuster-trained DeBERTa V2 (He et al., 2020) claim detector. We then verify the presence of these claims in the second answer through NLI. The similarity score is the average presence of factual claims across answers.

In QAMPARI, self-consistency-NER is used to determine confidence distribution. As the answers are entity lists, we extract entities by separating commas. The similarity score is calculated based on the proportion of overlapping entities between two answers, relative to the total entities in the first answer.

For ELI5, where answers provide easily understandable explanations, we focus on the overall content. Here, we apply a self-consistency-naive method, assign a similarity score to each answer pair with simple prompting.

In CNNDM, where answers are summaries highlighting key points of an article, we gauge the similarity between two answers by evaluating the overlap of key points. To achieve this, we implement the self-consistency-split method. We dissect the first answer into individual sentences and then use NLI to determine if each sentence is present in the second answer. The similarity score is derived by averaging the presence of these segmented sentences in the comparative answer.

E Improving Calibration

E.1 Fine-tuning

In our study, we focus on improving model calibration on the ASQA dataset. We explore three different strategies: fine-tune the model to do self-evaluation (Input: question and model's answer; Output: answer's score and explanation), fine-tune the model to do generation (Input: question; Output: answer), and a hybrid of both.

Data We generate a self-evaluation training dataset with GPT-4, which evaluates different models responses to questions drawn from ASQA training set. This dataset comprises inputs of questions and corresponding model answers, with outputs including a score and an explanation for each answer. We create the self-evaluation data in two steps. The initial phase involves the creation of a diverse answer pool. This is achieved by employing a suite of models with varying computational capacities, including Llama-2-7b, Llama-2-13b, Llama2-70b, Vicuna-13b, and ChatGPT. Each model generates responses to a spectrum of questions drawn from the training set of ASQA task, ensuring the resultant answer pool encompasses a broad quality spectrum, from low to high. Subsequently, we employ GPT-4 to critically assess these answers, assigning a score and providing a corresponding explanation for its evaluation. This process yields a rich dataset, each instance of which encompasses a question, a model-generated answer, an evaluative score, and a justification for that scoring. This approach results in a comprehensive dataset with 2,000 self-evaluation examples (1,800 for training and 200 for validation), each including a question, model-generated answer, score, and justification. For generation data, we use 80% of training ASQA's dataset (4,353 examples) for training and the remaining 20% for validation. The hybrid dataset combines the self-evaluation and generation training sets, using the self-evaluation validation set for assessment.

Training and Results Regarding training, we fine-tune Llama2-13b-chat model using LORA on this dataset. We maintain consistent parameters across all scenarios: a learning rate of 5e-6, five epochs, 100 warm-up steps, and a total batch size of 4 (achieved through 4 gradient accumulation steps across four GPUs, with a batch size of 1 per device). As Table 3 reveals, solely training on self-evaluation ('Evaluation') did not yield consistent improvements in calibration, possibly due to the complexity of this task, as well as the limitation of LORA fine-tuning. Nonetheless, fine-tuning the model improves the self-consistency method, especially when the generation data is included during training ('Eval + Gen' and 'Generation'). The model becomes more confident in terms of self-consistency after fine-tuning.

E.2 Source Documents

We investigate the effect of additional context on LLM calibration, focusing on self-consistency confidence with Llama-2-13b-chat and GPT-3.5-turbo models on the ASQA dataset. We test the model's performance when supplemented with two different types of source documents: random documents related to the question and 'oracle' documents directly relevant to the answers, as included in the dataset release (Stelmakh et al., 2022). Findings in Table 5 reveal that Oracle documents can enhance model performance and calibration across two out of three metrics for both models, while random documents are less effective. These results underscore the importance of relevant contextual information in model calibration.

Model	**Doc**	**Corr**	**ECE-M**	**F1$_{0.8}$**	**Score**
Llama2$_{13b}$	N	48.0	15.9	3.6	51.3
	R	39.0 ↓	14.8↓	30.6 ↑	59.6 ↑
	O	49.5 ↑	17.5↑	20.1 ↑	65.8 ↑
GPT-3.5$_{turbo}$	N	30.5	26.7	58.4	72.6
	R	36.6 ↑	26.4↓	68.2 ↑	64.8 ↓
	O	27.0 ↓	24.9↓	72.8 ↑	75.8 ↑

Table 5: How source documents affect the calibration score. In the document column, "N" means no documents in the input prompt, "R" means randomly selected documents relevant to the topic, and "O" means the oracle documents relevant to the answer. ↑ denotes that the calibration score goes up when adding documents, while ↓ means going down. For ECE-M it's opposite.

E.3 Hybrid Confidence Elicitation

We explore whether combining self-evaluation and self-consistency can yield a more accurate confidence distribution on ASQA dataset. By blending confidence distributions from self-evaluation (C_i^{eval}) and self-consistency ($C_i^{consist}$) into a hybrid distribution $C_i^{hybrid} = \alpha C_i^{eval} + (1 - \alpha) C_i^{consist}$, we adjust their relative contributions using α. As shown in Figure 13, we observe that the correlation between confidence and correctness initially increases but then declines as α varies from 0 to 1. However, this trend doesn't extend to other metrics like F1, indicating that while hybrid calibration elicitation may enhance calibration in terms of correlation, it may not have the same impact on other dimensions.

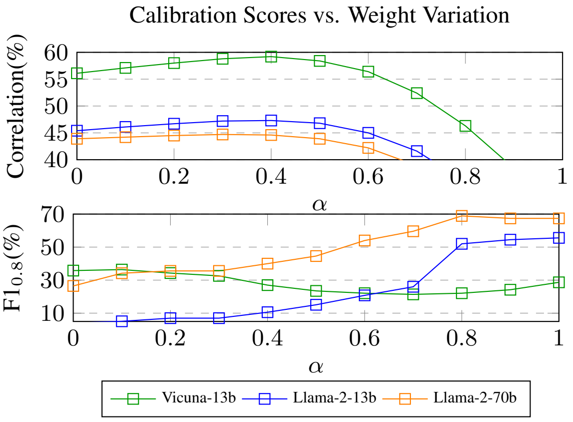


Figure 13: Hybrid confidence elicitation

F Prompts

This section introduces the prompts used for our experiments.

F.1 Correctness Evaluation Template

Similar to contemporary work (Kim et al., 2023), our evaluation template for GPT-4 evaluation to get the target correctness distribution of an answer includes four components: a clear task description, expertly crafted evaluation criteria for objectivity, demonstrations with a variety of answer qualities (best, worst, intermediate) each with a score and rationale, and specific evaluation instructions for the LLM, encompassing the question-answer pair to be evaluated and a reference answer.

Evaluation Template

{task instruction}

You will be given a question, a reference answer, and a student's answer. Please evaluate the student's answer based on both your knowledge and the reference answer, and provide a score from 0-5 to the student's answer. Keep in mind that the reference answer is not the sole correct response. Assess for both factual accuracy and relevance to the question. The following are the scoring criterion:

{criterion}

Here are some examples.

{examples}

Now it's your turn.

Question: {question}

Reference answer: {reference answer}

Student's answer: {answer}

Now please provide your score about this answer in the format of "Score: <Your score>/5" and give your explanation.

F.2 Criteria

Below are the criteria for various tasks, with a special note that the CNNDM summarization task utilizes a distinct evaluation template.

ASQA Criterion

ASQA Criterion

5 - Completely Correct and Highly Relevant: The answer fully addresses the question, resolves the ambiguity, and provides a well-rounded resolution. All facts presented in the answer are accurate and relevant.
4 - Mostly Correct and Relevant: The answer is very relevant and addresses the ambiguity well, but might have a minor oversight or inaccuracy. All the facts presented are accurate and relevant, or with only minor errors.
3 - Partially Correct and Relevant: The answer is generally on topic and attempts to address the ambiguity, but there might be inaccuracies or omissions. The majority of the facts are correct, with a few errors.
2 - Flawed but Somewhat Relevant: The answer somewhat addresses the topic but does not fully explore the question's ambiguity or does not provide a complete resolution. The facts presented are a mix of correct and incorrect information, with about half being accurate.
1 - Mostly Incorrect or Mostly Irrelevant: The answer slightly touches upon the topic but misses the main point. The majority of the facts presented are incorrect, with only a small portion being accurate.
0 - Completely Incorrect or Completely Irrelevant: The student's answer is completely off-topic, not related to the question at all, or contains only incorrect information.

ELI5 Criterion

ELI5 Criterion

5 - Perfectly Addressed, Accurate and Clarity: The answer flawlessly addresses the question with exceptional accuracy and clarity. It simplifies complex concepts effectively and does so in a way that is captivating and memorable.
4 - Accurate and clear: The answer is accurate, relevant to the question, and presented in a way that is engaging and understandable. It simplifies complex concepts effectively but may miss a small opportunity for further clarification or engagement.
3 - Moderately Accurate and Understandable: The answer is mostly accurate and somewhat understandable. It addresses the question reasonably well but may lack detail or contain some inaccuracies. It may use complex terms or concepts that are not broken down into simpler ideas.
2 - Relevant but Lacks Clarity or Accuracy: The answer is related to the question but lacks clarity or contains partial inaccuracies. It attempts to simplify the idea but does not do so effectively, leaving room for confusion or misunderstanding.
1 - Significantly Flawed: The answer addresses the question to a minimal extent but contains significant inaccuracies or misleading information. It might show a basic attempt to simplify the concept but fail in accuracy or relevance.
0 - Completely Inaccurate or Irrelevant: The answer is entirely off-topic, irrelevant, or factually incorrect. It fails to address the question and does not simplify complex ideas.

QAMPARI Criterion

* Estimated Accuracy Assessment (0-3 Points)
* 3 Points: All answers provided seem correct based on available knowledge.
* 2 Points: Majority of the answers seem correct.
* 1 Point: Some answers are correct, but there are notable inaccuracies.
* 0 Points: No answers seem correct or very high degree of inaccuracy.

* Estimated Completeness Assessment (0-2 Points)
* 2 Points: The response seems comprehensive, covering a broad range of known or expected correct answers.
* 1 Point: The response covers some correct answers but misses significant known or expected answers.
* 0 Points: The response is highly incomplete, missing most of the known or expected correct answers.

* Total Score (0-5 Scale)
* Add the points from Estimated Accuracy and Estimated Completeness.

Summarization Evaluation Template

You will be given one summary written for a news article.

Your task is to rate the overall quality of the summary with a score from 0 to 5, where 0 is the lowest and 5 is the highest.

Please make sure you read and understand these instructions carefully. Please keep this document open while reviewing, and refer to it as needed.

Evaluation Steps:
1. Read the news article carefully and identify the main topic and key points.
2. Read the summary and compare it to the news article. Check if the summary covers the main topic and key points of the news article, and if it presents them in a clear and logical order.
3. Assign a score for the summary quality on a scale of 0 to 5, where 0 is the lowest and 5 is the highest.

Task Input:

Article: {article}

Summary: {summary}

Now please provide your score of the summary in the format of "Score: <Your score>/5" and give your explanation.

F.3 Self-Evaluation Template

Self Evaluation Template

{task instruction}

You will be given a question and a student's answer. Please evaluate the student's answer, and provide a score from 0-5 to the student's answer based on the following scoring criterion:

{criterion}

Here are some examples.

{examples}

Now it's your turn.

Question: {question}

Answer: {answer}

Now please provide your score about this answer in the format of "Score: <Your score>/5" and give your explanation. Assess for both factual accuracy and relevance to the question.

F.4 Self-Consistency Template

In self-consistency approaches, assessing the similarity between two answers requires the use of an additional LLM in naive and split/claim methods. For the self-consistency-naive method, this involves an LLM directly providing similarity scores for the pair of answers. Meanwhile, for the self-consistency-split/claim method, another LLM is tasked with detecting the presence of sentences from answer 1 in answer 2. The templates for both the naive method and the split/claim method are outlined below.

Similarity Comparison Naive Template

You will be presented with a question followed by two answers. Evaluate how similar these answers are, considering their amount of information provided, factual content, effectiveness in addressing the question, format and organization. Conclude by providing an integer similarity score between 0 and 5.
Question: {question}
Answer 1: {answer1}
Answer 2: {answer2}
Now please give your similarity score in the format of "Similarity score: <Your score>/5" and give your explanation. Make your answer short and concise.

Similarity Comparison Split Template

You will receive a sentence and a response; please ascertain if a similar statement is present in the response.
Sentence: {sentence}
Response: {response}
Verify if the response contains a statement resembling the target sentence. Answer yes or no.

G Qualitative Analysis

We showcase successful and failed calibration examples on Qampari and ASQA with self-consistency method.

type	content
QAMPARI Successful Example 1	
Question	Who directed a film that had P. Balachandran as a screenwriter?
Answer	Kamal, P. Balachandran, T. K. Rajeev Kumar, V. K. Prakash, Rajeev Ravi.
Model Answer	Adu, Ammayum Muyalum, Bhoothakkannadi, Chattambi, Kakkakku Pinnaka...
Other Answer 1	Agraharathile Kazhuthai, Kaliyugam, Kodungallur, Kodiyettam ...
Other Answer 2	Kodiyettam, Kuttyedam, Oru CBI Diary Kurippu...
Other Answer 3	Aravindan, K.G. George, John Abraham, Shyam Benegal ...
...	...
Self-consistency scores	[0, 0, 0, ...] (%)
Correctness	Score: 0% Distribution: $\{0 : 1.0\}$
Confidence score	Score: 0% Distribution: $\{0 : 1.0\}$
QAMPARI Successful Example 2	
Question	Which cities or municipalities are located in the province of Basilan?
Answer	Akbar, Al-Barka, Hadji Muhtamad, Isabela City, Lamitan, Lantawan, Maluso, Sumisip, Tabuan-Lasa, Tipo-Tipo, Tuburan, Ungkaya Pukan.
Model Answer	Al-Barka, Basilan, Isabela, Lamitan, Tipo-Tipo, Maluso, Sumisip.
Other Answer 1	Basilan, Isabela, Lamitan, Lantawan, Tipo-Tipo, Sumisip.
Other Answer 2	Basilan, Tipo-Tipo, Isabela City, Maluso, Sumisip, Tuburan.
Other Answer 3	Basilan, Isabela, Tipo-Tipo, Lamitan, Sumisip, Al-Barka, Maluso, Tuburan.
...	...
Self-consistency scores	[71.4, 57.1, 85.7, ...](%)
Correctness	Score: 83.3% Distribution: $\{0.8 : 1.0\}$
Confidence	Score: 76.2% Distribution: $\{0.6 : 0.11, 0.8 : 0.78, 1.0 : 0.11\}$
QAMPARI Failed Example 1 (Under Confident)	
Question	What city or municipality located in the Philipines are in the Tarlac province?
Answer	Anao, Bamban, Camiling, Capas, Concepcion, Gerona, La Paz, Mayantoc, Moncada, Paniqui, Pura, Ramos, San Clemente, San Jose, San Manuel, Santa Ignacia, Tarlac City, Victoria.
Model Answer	Capas, Concepcion, Gerona, La Paz, Mayantoc, Paniqui, San Jose, Santa Ignacia, Tarlac City, Victoria.
Other Answer 1	Capas, Concepcion, Camiling, Tarlac City, Paniqui, San Jose, Santa Lucia, Victoria.
Other Answer 2	Capas, Tarlac, Concepcion, Tarlac, Gerona, Tarlac, La Paz, Tarlac, Paniqui, Tarlac, San Jose, Tarlac, Tarlac City.
Other Answer 3	Capas, Concepcion, Camiling, Tarlac City, San Jose.
...	...
Self-consistency scores	[60, 60, 30, ...](%)
Correctness	Score: 100% Distribution: $\{1 : 1.0\}$
Confidence	Score: 51.1% Distribution: $\{0.4 : 0.56, 0.6 : 0.33, 0.8 : 0.11\}$
QAMPARI Failed Example 1 (Over Confident)	
Question	What are some notables mailing list softwares ?
Answer	Dada Mail, Discourse, GNU Mailman, GroupServer, LISTSERV, Majordomo, Organic Groups Mailing List, Sympa.
Model Answer	Mailchimp, Constant Contact, Campaign Monitor, AWeber, GetResponse, MailPoet, Sendinblue, MailerLite, Klaviyo, ConvertKit.
Other Answer 1	Mailchimp, AWeber, GetResponse, Campaign Monitor, Constant Contact, Klaviyo, ConvertKit, MailPoet, Mailster, Sendinblue, MailerLite.
Other Answer 2	Mailchimp, Constant Contact, AWeber, Campaign Monitor, GetResponse, Klaviyo, MailPoet, Mailster, Sendinblue, MailerLite, ConvertKit, ActiveCampaign, and Constant Contact.
Other Answer 3	Mailchimp, AWeber, GetResponse, Campaign Monitor, MailerLite, Constant Contact, Klaviyo, ConvertKit, Sendinblue, and Emma.
...	...
Self-consistency scores	[100, 100, 90, ...](%)
Correctness	Score: 0% Distribution: $\{0 : 1.0\}$
Confidence score	Score: 97.8% Distribution: $\{0.8 : 0.22, 1.0 : 0.78\}$

Table 6: Examples for QAMPARI

type	content
	ASQA Successful Example 1
Question	What are the three types of enzymes in digestion?
Answer	Digestive enzymes are a group of enzymes that break down polymeric... Lipases split fatty acids... Proteases and peptidases split proteins ... Amylases split carbohydrates ... Nucleases split nucleic acids into nucleotides...
Model Answer	There are three main types of enzymes involved in digestion: proteases, which break down proteins; amylases, which break down carbohydrates; and lipases, which break down fats. Proteases are produced by the stomach and pancreas, while amylases are ... These enzymes work in concert ...
Other Answer 1	Digestive enzymes are proteins that help ... There are three main types of digestive enzymes: proteases, which break down proteins into amino acids; amylases, which break down carbohydrates into simple sugars; and lipases, which break down fats into glycerol and fatty acids. These enzymes are produced by different organs and ...
Other Answer 2	There are three main types of enzymes involved in digestion: proteases, which break down proteins; amylases, which break down carbohydrates; and lipases, which break down fats. These enzymes are produced by the salivary glands, stomach, and pancreas, and work in various parts of the digestive system...
...	...
Self-consistency scores	[100, 100, 0, ...] (%)
Correctness	Score: 100% Distribution: $\{1.0 : 1.0\}$
Confidence score	Score: 95.6% Distribution: $\{0.8 : 0.22, 1.0 : 0.78\}$
	ASQA Successful Example 2
Question	When did ben rectors new album come out?
Answer	Benjamin Evans Rector is an American singer... He has released seven studio albums, including Magic on June 22, 2018, which peaked at No. 1 .. Brand New which released on August 28, 2015... Older albums such as Something Like This, released on September 13, 2011 and The Walking in Between, came out August 20, 2013.
Model Answer	Ben Rector's new album, "Magic", was released on March 25, 2022.
Other Answer 1	Ben Rector's new album, "Brand New", was released on March 10th, 2022.
Other Answer 2	Ben Rector, an American singer-songwriter, released his new album "Magic" on October 20...
...	...
Self-consistency scores	[0, 0, ...](%)
Correctness	Score: 0% Distribution: $\{0 : 1.0\}$
Confidence	Score: 0% Distribution: $\{0 : 1.0\}$
	ASQA Failed Example 1 (Under Confident)
Question	Who plays charlie on it's always sunny?
Answer	Charlie Day plays the fictional character Charlie Kelly... Day is an American actor, screenwriter...
Model Answer	Charlie Day plays Charlie on It's Always Sunny in Philadelphia. He is also known for his role as Dale Arbus ... Day is a member of the improvisational comedy ...
Other Answer 1	Dennis is played by Glenn Howerton.
Other Answer 2	Charlie on It's Always Sunny in Philadelphia is played by Charlie Day.
...	...
Self-consistency scores	[0, 33.3, ...](%)
Correctness	Score: 80% Distribution: $\{0.8 : 1.0\}$
Confidence	Score: 31.1% Distribution: $\{0 : 0.22, 0.4 : 0.78\}$
	ASQA Failed Example 2 (Over Confident)
Question	Who sings wake me up when it's all over?
Answer	"Wake Me Up" is a song by Swedish DJ and record producer Avicii. Aloe Blacc, which is the stage name for Egbert Nathaniel Dawkins III provides vocals for the song while Mike Einziger provides acoustic guitar. "Wake Me When It's Over" is a song written and recorded by Willie...
Model Answer	"Wake Me Up When September Ends" is a song written by Billy
Other Answer 1	"Wake Me Up When September Ends" is a song by American rock band Green Day, written by lead vocalist Billie Joe Armstrong...
Other Answer 2	"Wake Me Up When September Ends" is a song by Green Day, written by the band's lead vocalist and guitarist Billie Joe Armstrong...
...	...
Self-consistency scores	[100, 100, ...](%)
Correctness	Score: 0% Distribution: $\{0 : 1.0\}$
Confidence score	Score: 77.8% Distribution: $\{0 : 0.22, 1.0 : 0.78\}$

Table 7: Examples for ASQA