Evaluating and Calibrating LLM Confidence on Questions with Multiple Correct Answers

Anonymous ACL submission

Abstract

Confidence calibration is essential for making large language models (LLMs) reliable, yet existing training-free methods have been primarily studied under single-answer question answering. In this paper, we show that these methods break down in the presence of multiple valid answers, where disagreement among equally correct responses leads to systematic underestimation of confidence. To enable a systematic study of this phenomenon, we introduce MACE, a benchmark of 12,000 factual questions spanning six domains with varying numbers of correct answers. Experiments across 15 representative calibration methods and four LLM families (7B–72B) reveal that while accuracy increases with answer cardinality, estimated confidence consistently decreases, causing severe miscalibration for questions with mixed answer counts. Notably, larger models are more severely affected, as they internalize substantially more knowledge. To address this issue, we propose Semantic Confidence Aggregation (SCA), which aggregates confidence over multiple high-probability sampled responses. SCA achieves state-of-the-art calibration performance under mixed-answer settings while preserving strong calibration on single-answer questions.

1 Introduction

Confidence calibration aims to align a model's expressed confidence with its true predictive performance (Guo et al., 2017; Desai and Durrett, 2020), and has long been a central research topic for improving the reliability of machine learning systems. In recent years, confidence calibration for large language models (LLMs) has attracted increasing attention, as honesty has been recognized as a fundamental aspect of aligning LLMs with human values (Askell et al., 2021). When properly calibrated, LLMs' confidence can serve as a meaningful indicator of their uncertainty, reflecting the likelihood of hallucination (Maynez et al., 2020; Min et al., 2023) and guiding when retrieval augmentation is required (Ni et al., 2024).

Confidence calibration methods can be divided into training-based and training-free approaches. Training-based methods learn to align confidence with ground-truth accuracy and can achieve strong in-domain calibration (Lin et al., 2022; Ni et al., 2025b). However, they require substantial annotation and often generalize poorly to other questions or domains (Ni et al., 2025a). In this paper, we focus on training-free methods, which can be roughly categorized into four approaches: 1) leveraging token-level generation probabilities (Guo et al., 2017; Desai and Durrett, 2020; Jiang et al., 2021; Si et al., 2022), 2) prompting models to explicitly verbalize confidence (Lin et al., 2022; Yin et al., 2023; Xiong et al., 2023; Ni et al., 2024), 3) measuring semantic consistency across multiple sampled responses (Manakul et al., 2023; Kuhn et al., 2023; Zhang et al., 2023), and 4) employing the model itself to evaluate whether its generated answer is correct (Kadavath et al., 2022; Tian et al., 2023).

Among these, response-consistency–based methods achieve state-of-the-art (SOTA) performance. The underlying assumption is that greater diversity among sampled responses signals higher model uncertainty—if the model "knows" the answer, its outputs should converge. This assumption is reasonable when each question admits a single correct answer, as divergent outputs genuinely reflect the model's hesitation. However, this assumption breaks down when a question has multiple valid answers. As illustrated in Figure 1, consider two questions that both yield 18 out of 20 correct sampled responses. For the single-answer question "What is the active ingredient in Tylenol for headaches?" (panel a), the 18 correct responses all converge on the same answer—Acetaminophen—and the consistency-based confidence estimate is a trustworthy 0.9. For

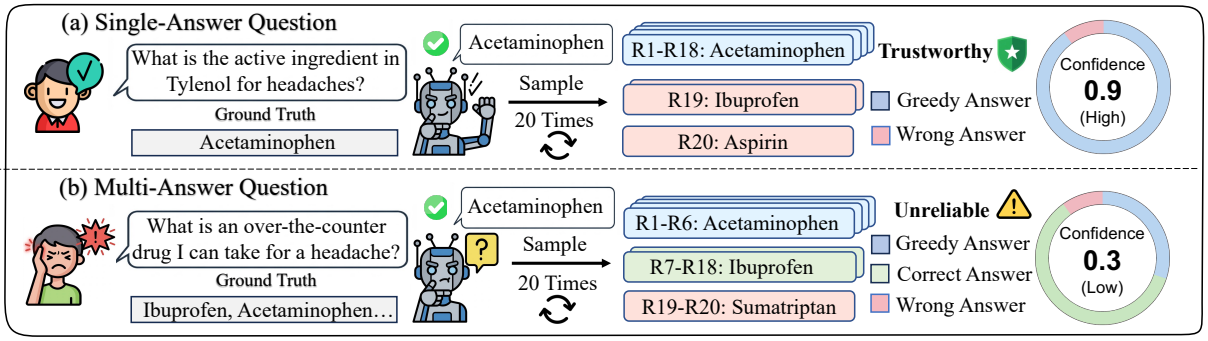

Figure 1: **Failure of consistency-based calibration on multi-answer questions.** Disagreement among sampled correct answers yields low estimated confidence, indistinguishable from the low confidence caused by the presence of incorrect answers, thereby making correct responses appear unreliable.

the multi-answer question "What is an over-the-counter drug I can take for a headache?" (panel b), however, the 18 correct responses split across equally valid answers—Acetaminophen (R1–R6) and Ibuprofen (R7–R18). Despite achieving the same accuracy, this natural variation among correct answers drives the estimated confidence down to 0.3, and the model is flagged as unreliable. The model suffers a severe trust crisis not because it is wrong, but because it is right in more than one way.

Existing benchmarks largely overlook the multi-answer setting and are not directly suitable for evaluating confidence calibration under it (see Appendix A for a detailed discussion). To systematically study confidence estimation under multi-answer question answering (QA), we introduce MACE (**M**ulti-**A**nswer **C**onfidence **E**stimation), a benchmark of 12,000 question–answer pairs spanning six factual domains, where each question has 1, 2, 4, or 6 correct answers.

Using MACE, we evaluate 15 representative confidence calibration methods across four LLM families (LLaMA, Qwen, DeepSeek, and GPT), ranging from 7B to 72B parameters. Our experiments reveal that: (1) as the number of correct answers increases, LLMs achieve higher QA accuracy but exhibit lower confidence; (2) in realistic settings with mixed ground-truth answer cardinalities, methods that achieve SOTA calibration on single-answer questions collapse due to severe miscalibration on multi-answer questions; and (3) this confidence degradation is more pronounced for larger models (e.g., 70B), as they tend to alternate among a wider set of correct answers than smaller models.

To address this, we propose SCA (**S**emantic **C**onfidence **A**ggregation), which calibrates confidence by aggregating the probabilities of multiple high-confidence answers rather than relying on the most confident one. Specifically, we use the token-level sequence generation probability of each sampled response as its confidence signal. Since low-confidence responses contribute minimally, a simple summation over sampled responses yields improved calibration under mixed-answer settings, outperforming SOTA baselines while preserving strong performance on single-answer questions.

In summary, this work advances confidence calibration from single-answer QA to a more general setting with mixed numbers of ground-truth answers. We introduce the MACE benchmark, conduct a comprehensive evaluation across four LLM families, and propose SCA, which achieves state-of-the-art calibration performance in this general QA scenario.

2 Related Work

Calibration in Large Language Models. Existing calibration methods for large language models (LLMs) can be training-based and training-free. Training-based methods require finetuning to enhance knowledge boundary perception (Lin et al., 2022; Zhang et al., 2024; Yang et al., 2024b). Training-free methods can be broadly categorized into four approaches: *Probability-based* methods leverage the model's internal representations, such as token probabilities, logits, or hidden activations (Jiang et al., 2021; Kadavath et al., 2022; Ren et al., 2022; Wightman et al., 2023; Ling et al., 2024; Ni et al., 2025b). *Verbalized-based* methods prompt LLMs to directly express confidence in natural language (Lin et al., 2022; Tian et al., 2023; Tanneru et al., 2024). *Consistency-based* methods estimate reliability by measuring agreement across multiple generations or reasoning chains (Manakul et al.,

2023; Cole et al., 2023; Lyu et al., 2025; Ni et al., 2025a). *Self-evaluation-based* methods ask the model to judge its own correctness (Kadavath et al., 2022; Tian et al., 2023).

Datasets for Confidence Estimation. Existing benchmarks can be categorized by answer-space structure. *Multiple-choice* benchmarks include MMLU-PRO (Wang et al., 2024), ARC (Clark et al., 2018), and SATA-Bench (Xu et al., 2025). *Open single-answer QA* benchmarks include TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and Mintaka (Sen et al., 2022). *Open multi-answer QA* benchmarks include WebQuestions (Berant et al., 2013) and ComplexWebQuestions (Talmor and Berant, 2018). However, none simultaneously provides exhaustive ground truth, multi-answer questions, and controlled answer cardinality—three properties essential for studying calibration under probability mass split (see Appendix A for a detailed analysis). To fill this gap, we introduce the benchmark **MACE**.

3 The MACE Benchmark

In this section, we provide the overview and construction process of the MACE benchmark.

3.1 Overview

To study how existing confidence calibration methods perform on multi-answer questions, we introduce MACE (**M**ulti-**A**nswer **C**onfidence **E**stimation), a benchmark comprising questions with one or multiple correct answers. MACE spans six factual domains: Honorable Award (Award), Political Office (Office), Regional Affiliation (Region), Mathematical Concept (Math), Natural River (River), and Linguistic Culture (Language). For each domain, MACE includes four question types with 1, 2, 4, or 6 correct answers, respectively (abbreviated as 1a, 2a, 4a, and 6a, in the paper), with 500 questions per type. The construction of MACE has three key phases, shown in Figure 2: **knowledge collection** that collects knowledge triplets $\{\text{subject}, \text{relation}, \text{object}\}$ in the domains, **knowledge filtering** that filters out low-quality triplets, and **QA pair generation** that constructs natural language QA pairs based on the triplets. Dataset statistics are summarized in Table 1.

3.2 Knowledge Collection

Domain Identification. The challenge in data construction lies in identifying questions that allow multiple valid answers while remaining complete. For example, for the question "Tell me a great scientist in 2025," it is impractical to enumerate all correct answers. To ensure that answers are clear and complete, we begin by collaborating with LLMs to identify suitable domains. Balancing coverage and annotation cost, we ultimately select six domains (See Table 1). Details are provided in Appendix B.1.

Table 1: **Statistics of the six factual domains in MACE. Subj.**: number of subjects collected from Wikidata; **Obj.**: number of distinct objects associated with the subjects; **Const-Q**: number of questions constructed from subject–object pairs; **Used-Q**: number of used questions via clustering-based stratified sampling.

Domain	Subj.	Obj.	Const-Q	Used-Q
Award	175	1,634	9,384	2,000
Office	159	1,706	6,177	2,000
Region	99	1,074	5,120	2,000
Math	5	2,957	10,000	2,000
River	781	114	10,000	2,000
Language	847	141	10,000	2,000

Triplet Collection. In this step, we collect knowledge for each domain. For the Math domain, knowledge is derived via predefined knowledge types, such as identifying prime numbers. For other domains, following prior research (Sen et al., 2022; Yuksekgonul et al., 2023; Mallen et al., 2023), we collect knowledge triplets from Wikidata[1] which is a widely recognized knowledge source. Knowledge triplets are structured in the form of (subject, relation, object). For a given domain, we first retrieve all relevant subjects, and then query the corresponding objects. The subject type, relation, and object type of each domain can be found in Table 6. The subject counts are reported in the `Raw` column of Table 5. More details are described in § B.2.

3.3 Knowledge Filtering

To ensure answer correctness and completeness, we conduct heuristic filtering and manual verification. The count of the remaining subjects after each step is shown in Table 5.

We use a two-stage pipeline to remove low-quality data: (i) *Popularity Filter*: retains only high-traffic subjects based on Wikipedia pageviews; (ii) *Validity Filter*: further eliminates invalid or noisy subjects via domain-specific rules. Details can be found in Appendix B.3. This step

[1] https://query.wikidata.org/sparql

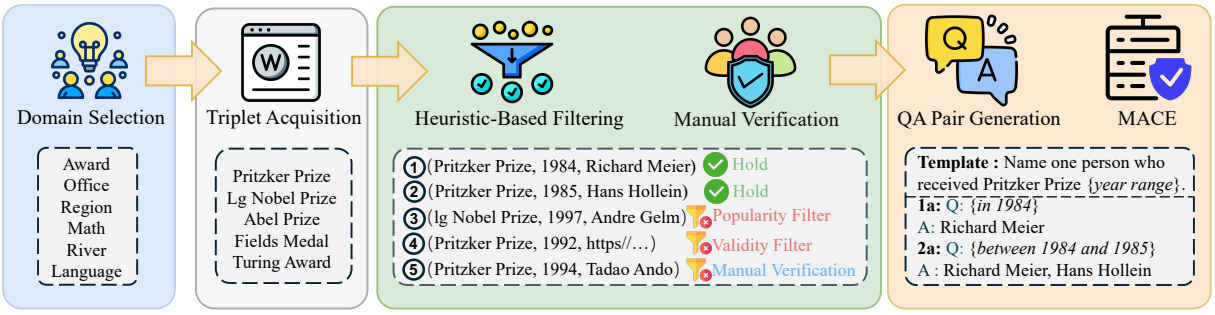

Figure 2: **The MACE benchmark construction pipeline**, illustrated using the *Award* domain. ①② **Valid triplets** (e.g., Pritzker Prize winners) are retained. ③ **Low-popularity triplets** (e.g., obscure awards like the Ig Nobel Prize) are removed via *Popularity Filter*. ④ **Noisy triplets** containing invalid formats (e.g., URLs) are removed via a *Validity Filter*. ⑤ **Factually incorrect triplets** are identified and removed during the *Manual Verification*. Finally, QA pairs are generated using triplets under four ground-truth counts settings (1a, 2a, 4a, 6a).

filters 92.4% of the data, retaining a high-quality subset and substantially reducing the workload for human verification.

Manual Verification. We recruit ten domain experts to manually audit the remaining triplets, recognizing any remaining factual errors and enriching incomplete entries. Finally, this step removes an additional 19.05% of subjects, achieving high inter-annotator agreement (Cohen's $\kappa = 0.94$). Details can be found in § B.4.

3.4 QA Pair Generation

Following previous research (Mallen et al., 2023; Yuksekgonul et al., 2023), we construct QA pairs using the collected knowledge triplets. For each domain, we create questions based on a subject and its associated relation, with the object serving as the answer. By varying the subjects and relations, we can generate questions with different numbers of correct answers, e.g., "Name one person who received {award} between {lower_year} and {upper_year}". Details can be found in § B.5. The number of constructed questions is shown in the `Const-Q` column of Table 1. To prevent any domain from dominating the evaluation, we randomly sample 500 questions for each setting (1a, 2a, 4a, and 6a) in each domain for evaluation, resulting in 2,000 QA pairs per domain.

3.5 Construction Characteristics

Our construction pipeline ensures that: 1) all questions in MACE have clear, complete, and correct ground-truth answers; 2) questions with different numbers of valid answers are explicitly separated and sufficiently represented, enabling analysis of answer-count effects and realistic mixed-question scenarios; and 3) questions across varying answer-count settings are grounded in the same knowledge types and generated from the same prompt templates, differing only in factors such as year range, thereby maintaining comparable difficulty.

4 Experimental Setup

In this section, we describe our experimental settings, covering the categorization of confidence estimation methods, models, and evaluation.

4.1 Confidence Estimation Methods

We extensively evaluate 15 existing confidence estimation methods and broadly categorize them into single-turn and double-turn methods.

Single-turn. This class of methods estimates confidence along with answer generation and can be further divided into two types.

- **Question-level.** These methods quantify the entropy of the model's output distribution. Prediction Entropy *(Prob Entropy)* (Malinin and Gales, 2020) directly measures the entropy of the answer probability space; Length-Normalize Prediction Entropy*(N-Prob Entropy)* (Kadavath et al., 2022) refines this by normalizing for sequence length; and Semantic Entropy *(Sem Entropy)* (Kuhn et al., 2023) focuses on the semantic meaning regardless of the specific formatting.
- **Answer-level Confidence**. These methods evaluate the confidence on a specific answer. *Verb* prompts the model to directly express confidence in natural language, while *Verb-Topk* (Jiang et al., 2021) outputs confidence for top-k candidates. Consistency*(Consis)* (Manakul et al., 2023) measures confidence via agreement across multiple

sampled outputs. *Consis-Verb* and *Consis-Verb-Topk* (Xiong et al., 2023) further weight this consistency using the verbalized scores from *Verb* and *Verb-Topk*. *Perplexity* derives confidence from the sequence's generation probability.

Double-turn. These methods adopt a post-hoc approach, measuring confidence through a secondary query after the initial answer is produced.

- P_{True} (Kadavath et al., 2022). The model verifies its answer via a binary `True/False` judgment. Specifically, P_{True}*-Consis* approximates the confidence of `True` via black-box sampling, whereas P_{True}*-Prob* directly extracts the token's probability. Their conditioned variants, P_{True}*-Consis-Cand* and P_{True}*-Prob-Cand*, augment the verification prompt with multiple candidate answers to assist the judgment.
- **Self-Ask** (Tian et al., 2023). The model explicitly verbalizes a numerical confidence score in natural language. *Self-Ask* performs this assessment directly, while *Self-Ask-Cand* incorporates multiple candidate answers as assistance.

Detailed formulations of each method are provided in Appendix § C.1. Compared with double-turn methods, we speculate that single-turn methods could be more strongly influenced by the number of answers, as it is closely tied to the probability of generating those answers.

4.2 Models

We employ a diverse set of representative language models across different scales and series. For open-source models, we include Qwen2.5-Instruct series (7B, 14B, 32B, 72B) (Yang et al., 2024a), LLaMA3.1-Instruct series (8B, 70B) (Grattafiori et al., 2024), and DeepSeek-V3 (Liu et al., 2024). Additionally, we include two closed-source models GPT-4o-mini and GPT-4o (Hurst et al., 2024).

4.3 Evaluation

We use **Accuracy** to measure QA performance, where an answer is considered correct if it matches the ground-truth answer, and report **Confidence** to reflect each method's prediction. Following previous research (Kuhn et al., 2023; Ni et al., 2025a), we use **AUROC** (Area Under the Receiver Operating Characteristic Curve) (Hanley and McNeil, 1982) to measure confidence quality. This metric evaluates whether the confidence assigned to correctly answered samples is higher than that assigned to incorrectly answered samples. More details can be seen in § C.2. For each metric, we report the average across all domains. Per-domain results are provided in Appendix C.3.

Table 2: **Confidence scores of LLaMA-3.1-70B-Instruct** under different counts of correct answers. "1a" denotes questions with a single correct answer, and the remaining settings follow the same convention.

Method	**1a**	**2a**	**4a**	**6a**
Accuracy	48.0	55.4	59.9	61.7
Single-turn				
Question-level				
Prob Entropy	71.6	70.0	66.6	65.6
N-Prob Entropy	91.5	90.9	89.9	89.7
Sem Entropy	60.8	57.9	49.7	45.2
Answer-level				
Verb	94.8	94.7	93.1	92.3
Verb-Topk	81.9	73.9	62.1	56.9
Consis	51.3	48.9	40.2	35.9
Consis-Verb	50.3	47.9	40.5	36.4
Consis-Verb-Topk	47.6	45.9	38.4	34.2
Perplexity	77.6	76.4	73.0	71.0
Double-turn				
P_{True}-Consis	79.3	79.3	77.0	75.5
P_{True}-Prob	79.2	79.0	76.8	75.4
P_{True}-Consis-Cand	88.1	87.9	87.1	87.9
P_{True}-Prob-Cand	88.1	87.8	87.1	87.8
Self-Ask	63.6	60.9	56.1	54.4
Self-Ask-Cand	64.5	63.5	59.6	56.8

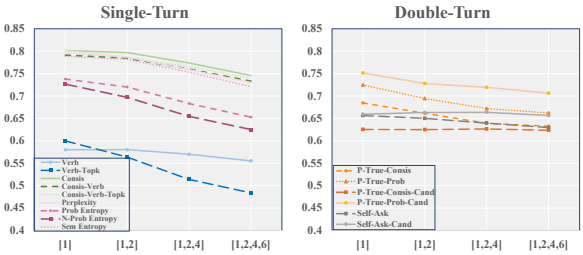

Figure 3: **AUROC of LLaMA-3.1-70B-Instruct** on mixed questions with varying counts of correct answers.

5 Results and Analysis

In this section, we examine how existing confidence estimation methods behave in multi-answer settings. We first analyze how QA performance and estimated confidence vary on questions with more correct answers, and then evaluate calibration performance under mixed questions with different answer counts. Due to space constraints, we focus on LLaMA-3.1-70B-Instruct, which shows trends consistent with other models; full results are provided in Appendix § D.1.

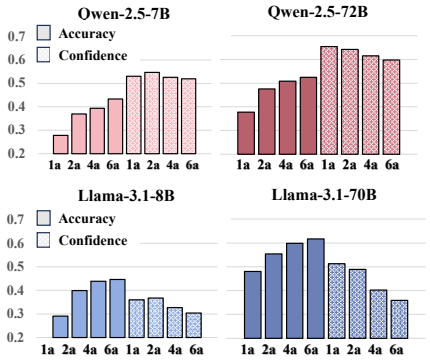

Figure 4: **QA performance and confidence variation** with the number of correct answers across model scales using *Consistency* method.

5.1 Model Confidence Decreases as the Number of Correct Answers Increases

As demonstrated in Table 2, increasing the number of ground-truth answers leads to a steady improvement in accuracy, suggesting that the questions become empirically easier. However, Table 2 reveals a consistent downward trend in confidence across most estimation methods. This indicates that miscalibration between confidence and QA performance arises in multi-answer settings for nearly all existing approaches. Across different domains, the overall trend is similar. In more challenging domains, such as Region, the model's confidence decreases less sharply (see Figure 8).

Larger models experience a sharper confidence drop. This phenomenon is shown in Figure 4. Within the LLaMA series, the 70B model exhibits a dramatic confidence drop of **0.154** from the single-answer (1a) to the six-answer (6a) setting, whereas the 8B model drops by only **0.056**—nearly three times smaller. We speculate that this is because larger models possess a much broader knowledge coverage. As the number of correct answers increases, larger models tend to generate a wider range of potential valid answers. In contrast, smaller models have limited knowledge coverage, so their generation space is often confined to a narrow subset of the possible answers. We provide a granular investigation into the impact of knowledge coverage, with detailed findings presented in Section 5.3.

5.2 Calibration Degrades on Mixed Questions with Varying Answer Counts

To evaluate existing methods in a more realistic scenario where questions can have different numbers of valid answers, we mix the questions and examine overall calibration performance for each domain. AUROC results are illustrated in Figure 3. We observe a *consistent downward trend in AUROC as questions become more heterogeneous in the number of correct answers for most of the methods*. This stems from the misalignment between rising accuracy and declining confidence estimates.

Single-turn methods are more sensitive to the number of answers. Compared with double-turn methods, single-turn methods typically exhibit stronger calibration performance, except for verbalized approaches, due to models' inherent overconfidence (Kadavath et al., 2022). However, their confidence scores are more sensitive to variations in the number of valid answers, leading to a faster degradation in calibration performance as the answer space diversifies. By contrast, double-turn methods are more stable across answer-count settings because they decouple confidence estimation from answer generation and rely on a post-hoc judge to verify answers; however, their overall calibration performance remains relatively limited. A detailed analysis is provided in § D.2.

5.3 Why Does Confidence Decline More Sharply in Larger Models?

We speculate that the decline in model confidence as the number of answers increases is related to the model's knowledge coverage. For models with limited knowledge, even as the number of correct answers increases, the candidate generation space may not expand substantially, leading to a relatively small confidence drop.

Definition of Knowledge Coverage. We consider a model's knowledge as the set of answers to which it assigns reasonable (i.e., not overly low) confidence. Following (Kuhn et al., 2023), we sample model responses and cluster semantically consistent answers to obtain the candidate generation space. We then filter out clusters with very low generation probabilities using a threshold of $\tau = 0.1$, since such clusters may arise from randomness rather than the model's actual knowledge. We define knowledge coverage as the number of remaining high-confidence answer clusters. A detailed analysis of the cluster probability distribution and threshold selection is provided in Appendix § D.3.

Figure 5 illustrates how knowledge coverage scales with model size, and Figure 10 shows the

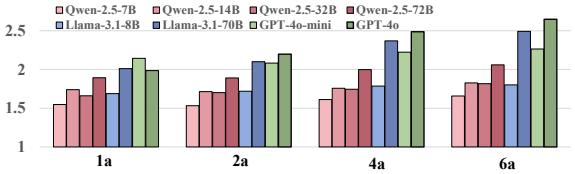

Figure 5: **Average knowledge coverage across ground-truth (GT) set sizes.** Different color schemes denote different model families. Color intensity increases to indicate increasing model size.

distribution of cluster sizes. Larger models consistently exhibit a greater knowledge coverage than smaller ones, with the gap widening as the number of ground-truth answers increases. This suggests that larger models encode broader knowledge, whereas smaller models may fail to produce effective answers even when more correct answers exist.

Consequently, as the number of correct answers grows, larger models—by virtue of their broader knowledge—tend to produce a more diverse set of plausible responses, which dilutes response consistency and thereby lowers estimated confidence. Smaller models, by contrast, draw from a narrower candidate pool, resulting in a more gradual confidence decline. This disparity arises because knowledge coverage expands more rapidly in larger models, amplifying the confidence drop. We further validate this relationship quantitatively in Appendix § D.4, where Spearman correlation analysis confirms a strong association between knowledge coverage and confidence degradation.

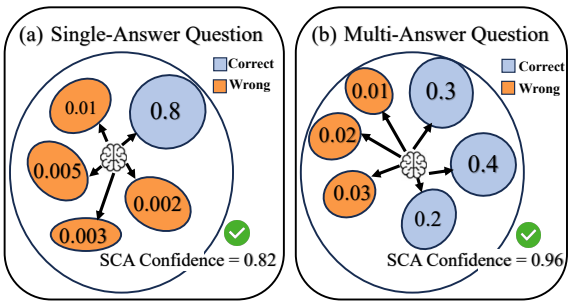

Figure 6: **Illustration of SCA method without tuning.** As shown, confidence estimation is dominated by high-confidence clusters. SCA sums probabilities across all clusters, achieving consistent strong calibration performance in both single-answer and multi-answer settings.

6 Semantic Confidence Aggregation

In this section, we propose Semantic Confidence Aggregation (SCA), a method for confidence calibration in realistic multi-answer scenarios. The key idea is that overall confidence should reflect the aggregate confidence of all valid answers rather than rely on the single most confident response, and that response inconsistency (e.g., high entropy) does not necessarily imply low confidence.

Table 3: **AUROC score on LLaMA-3.1-70B-Instruct** of different calibration methods under increasing answer mixture. $\tau = 0$ denotes no-filtering, while $\tau \neq 0$ uses the threshold value tuned on the development set. **Bold** marks the best, underlined the second-best in each column. † indicates the difference from our SCA method is not statistically significant ($p > 0.05$).

Method	[1]	[1,2]	[1,2,4]	[1,2,4,6]
Single-turn				
Question-level				
Prob Entropy	74.5	72.6	69.0	66.1
N-Prob Entropy	73.8	70.4	66.2	63.3
Sem Entropy	78.9	78.6	75.8	72.5
Answer-level				
Verb	58.4	58.4	57.2	55.8
Verb-Topk	60.7	56.2	51.2	48.4
Consis	80.2†	80.3	78.1	75.3
Consis-Verb	79.7†	79.0	76.6	73.8
Consis-Verb-Topk	80.0†	79.0	77.0	74.3
Perplexity	**80.9**†	79.4	76.5	73.8
Double-turn				
P_{True}-Consis	68.1	66.4	64.1	63.5
P_{True}-Prob	72.9	70.1	67.6	66.7
P_{True}-Consis-Cand	62.6	62.7	62.9	62.5
P_{True}-Prob-Cand	76.0	73.5	72.5	71.1
Self-Ask	66.5	66.1	65.0	63.9
Self-Ask-Cand	66.5	66.8	66.9	66.1
Confidence Aggregation Baselines				
SNCA ($\tau = 0.5$)	77.9	77.3	74.2	70.5
SNCA ($\tau = 0$)	50.0	50.0	50.0	50.0
SFCA ($\tau = 0.25$)	77.1	77.2	75.2	73.0
SFCA ($\tau = 0$)	50.0	50.0	50.0	50.0
Ours				
SCA ($\tau = 0.05$)	80.6	**81.2**	**79.2**	**76.7**
SCA ($\tau = 0$)	80.5	**81.2**	79.1	**76.7**

6.1 Methodology

Similar to Sem Entropy (Kuhn et al., 2023), we begin by sampling multiple answers and clustering them based on semantic similarity. Let $\mathcal{S} = \{s_1, \ldots, s_N\}$ denote N sampled responses for a given question. We partition $\mathcal{S}$ into M clusters $\mathcal{C} = \{C_1, \ldots, C_M\}$. Following Sem Entropy, for a cluster C_m, we compute its confidence as the sum of the token-level probabilities of all responses within the cluster. The formulation is:

$$P_{\text{SCA}}(C_m) = \sum_{s \in C_m} p(s), \qquad (1)$$

where $p(s)$ represents the token-level generation probability of sequence s.

Since low-probability responses may be generated by chance rather than representing answers the model considers likely correct, we sum only the probabilities of clusters whose confidence exceeds a threshold τ. The set of valid clusters is defined as $\mathcal{C}_{\text{valid}} = \{C_m \in \mathcal{C} \mid P(C_m) > \tau\}$ and the final confidence score for the question is computed as:

$$\text{Conf}_{\text{SCA}} = \sum_{C_m \in \mathcal{C}_{\text{valid}}} P(C_m). \quad (2)$$

Experimental results show that SCA achieves comparable performance at $\tau = 0$ to that obtained with the optimal τ (Table 3). *This indicates that simply summing token-level probabilities over all sampled responses is sufficient, without requiring clustering or filtering, thereby reducing computational cost.*

6.2 Experimental Setup

Additional Aggregation Methods. To verify whether using token-level probabilities to compute cluster confidence is appropriate, we compare two additional methods. These methods differ only in how cluster confidence is calculated.

- SNCA (**S**emantic **N**ormalized **C**onfidence **A**ggregation): normalizes the aggregated token-level probabilities across all clusters to capture relative magnitudes: $P_{\text{SNCA}}(C_m) = \frac{P_{\text{SCA}}(C_m)}{\sum_{j=1}^{M} P_{\text{SCA}}(C_j)}$.
- SFCA (**S**emantic **F**requency **C**onfidence **A**ggregation): defines probability of a cluster based on the count of samples within it without considering token-level probability: $P_{\text{SFCA}}(C_m) = \frac{|C_m|}{N}$.

Models and Datasets. Due to space constraints, experiments in this section focus on LLaMA-3.1-70B-Instruct; results for other models, which yield consistent conclusions, are reported in Appendix § E.1. We randomly sample 20% of MACE from each domain as a development set and use the remaining data for testing. For each aggregation method, two settings are evaluated: a fixed threshold $\tau = 0$ and an optimal threshold tuned on the development set with statistical significance tested.

6.3 Results and Analysis

Table 3 shows calibration performance on the test set, and Figure 7 shows how the three aggregation methods vary with the threshold on the development set. We observe that *SCA matches the best-performing existing methods in the single-answer setting and achieves the strongest results across all three mixed multi-answer scenarios with $\tau = 0$.* This means that SCA achieves strong calibration performance even without clustering or any filtering, simply by summing the token-level probabilities of all responses. Compared with SNCA and SFCA, SCA is less sensitive to the threshold. This indicates that the absolute values of token-level probabilities inherently capture the model's perception of answer correctness, assigning high probabilities to likely correct answers and low probabilities to spurious ones, as shown in Figure 6. Additionally, we evaluate on the Natural Questions (Kwiatkowski et al., 2019) dataset to test generalization; as shown in Table 11, SCA also achieves competitive performance on this single-answer benchmark.

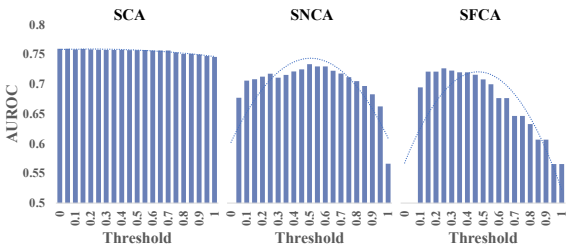

Figure 7: **Sensitivity of average AUROC across answer settings** to the threshold for aggregation methods.

7 Conclusion

This work extends confidence calibration from single-answer QA to a more general setting with mixed numbers of ground-truth answers. We introduce the MACE benchmark and evaluate existing calibration methods and find that, as the number of correct answers increases, LLMs achieve higher QA accuracy but lower estimated confidence, leading to severe miscalibration. Consequently, nearly all these methods collapse in realistic settings with mixed answer cardinalities. To address this, we propose SCA, which aggregates probabilities of multiple high-confidence answers rather than relying on the top one, achieving state-of-the-art calibration in this general multi-answer setting while preserving strong calibration on single-answer questions.

Limitations

While this study offers novel insights into confidence estimation within multi-answer settings, several limitations warrant further consideration. First, our benchmark employs formulaic prompt templates ("Name one X that satisfies Y") to ensure intra-domain difficulty homogeneity and controlled answer scaling (Appendix B.5). Although this design is a deliberate methodological choice to isolate confidence–rank dynamics, it sacrifices naturalistic diversity in question phrasing. Extending the benchmark with more heterogeneous surface forms—while preserving difficulty control—is an important direction for future work. Second, our evaluation is currently restricted to English-language datasets. Extending this analysis to multilingual or cross-lingual contexts remains essential for a more comprehensive understanding of calibration behavior across languages. Third, our scope primarily encompasses short factual responses. Since real-world applications often involve open-ended and compositional generation, investigating confidence behavior in long-form contexts is a crucial next step. Nonetheless, we believe our controlled evaluation framework provides a useful foundation for developing more robust calibration methods in realistic multi-answer scenarios.

Ethical Considerations

All data used in this study are derived from publicly available and legally accessible sources. We employ both open-weight and API-based large language models under their respective usage policies. All experiments are conducted strictly for academic research purposes, without involving human subjects or sensitive content.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, and 1 others. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

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

Jeremy Cole, Michael Zhang, Dan 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.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302.

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

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

James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a

benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, and 1 others. 2024. Uncertainty quantification for in-context learning of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3357–3370.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. 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.

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2025. Calibrating large language models with sample consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19260–19268.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 9802–9822.

Potsawee Manakul, Adian Liusie, and Mark John Francis Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *ArXiv*, abs/2303.08896.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1906–1919.

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.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11375–11388.

Shiyu Ni, Keping Bi, Jiafeng Guo, Minghao Tang, Jingtong Wu, Zengxin Han, and Xueqi Cheng. 2025a. Annotation-efficient universal honesty alignment. *arXiv preprint arXiv:2510.17509*.

Shiyu Ni, Keping Bi, Jiafeng Guo, Lulu Yu, Baolong Bi, and Xueqi Cheng. 2025b. Towards fully exploiting llm internal states to enhance knowledge boundary perception. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24315–24329.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2022. Out-of-distribution detection and selective generation for conditional language models. *arXiv preprint arXiv:2209.15558*.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. 2024. Quantifying uncertainty in natural language explanations of large language models. In *International Conference on Artificial Intelligence and Statistics*, pages 1072–1080. PMLR.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence

scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

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

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 preprint arXiv:2306.13063*.

Weijie Xu, Shixian Cui, Xi Fang, Chi Xue, Stephanie Eckman, and Chandan K Reddy. 2025. Sata-bench: Select all that apply benchmark for multiple choice questions. *arXiv preprint arXiv:2506.00643*.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024a. Qwen2.5 technical report. *ArXiv*, abs/2412.15115.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024b. Alignment for honesty. *Advances in Neural Information Processing Systems*, 37:63565–63598.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuan-Jing Huang. 2023. Do large language models know what they don't know? In *Findings of the association for Computational Linguistics: ACL 2023*, pages 8653–8665.

Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. 2023. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. *arXiv preprint arXiv:2309.15098*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024. R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7113–7139.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023. Sac3: reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458.

A Limitations of Existing Calibration Benchmarks

A well-calibrated language model should assign confidence scores that faithfully reflect its actual likelihood of being correct. Measuring this property in multi-answer factual QA, however, places stringent demands on the evaluation benchmark. In particular, we argue that three properties must be *jointly* satisfied (see Table 4 for a summary):

P1. Exhaustive Ground Truth. For every question, the ground-truth answer set must enumerate *all and only* the correct answers. Without exhaustiveness, a model that generates a valid but unlisted answer will be wrongly penalized as hallucinating, corrupting any calibration measurement.

P2. Multi-Answer Questions. The benchmark must contain questions whose correct answer set has cardinality $|\mathcal{A}| \geq 2$. Probability mass split—the phenomenon whereby a model's confidence in each individual correct answer is diluted because probability mass is dispersed across multiple valid completions—can only appear when multiple correct answers coexist. Single-answer benchmarks are structurally blind to this effect.

P3. Controlled Answer Cardinality. Questions should be organized into systematic cardinality tiers (e.g. $|\mathcal{A}| \in \{1, 2, 4, 6\}$) so that one can quantify how calibration degrades as the number of valid answers increases. Without such control, any observed miscalibration cannot be attributed to mass split with confidence.

We organize existing benchmarks along the dimension most relevant to probability mass split—*answer-space structure*—and show that none satisfies all three properties simultaneously. At the coarsest level, we distinguish benchmarks whose answer space is *closed* (a fixed set of options) from those requiring *open* free-form generation over the full vocabulary. Within the open category, we further separate benchmarks by whether correct answers are unique or multiple, and, in the latter case, whether the ground truth is exhaustive.

Table 4: Comparison of existing benchmarks with respect to three properties required for diagnosing *probability mass split* in multi-answer factual QA. **P1**: exhaustive ground truth; **P2**: multi-answer questions; **P3**: controlled answer cardinality. Categories are organized by *answer-space structure*.

Category	Representative	P1	P2	P3	Key Limitation
Closed answer space (fixed option sets)					
Single-correct MC	MMLU-PRO, ARC	✓	✗	✗	Fixed option sets bypass open-ended probability dispersion over the full vocabulary.
Multi-correct MC	SATA-Bench	✓	✓	✗	Correct-option count is not systematically controlled, precluding cardinality analysis.
Open answer space (free-form generation)					
Single-answer	TriviaQA, NQ, Mintaka	✓	✗	✗	Single-entity answers; exhaustiveness is trivially satisfied but no mass split can occur.
Multi-answer (non-exhaustive)	WebQuestions, CWQ	✗	✓	✗	Incomplete knowledge bases yield non-exhaustive ground truth, contaminating calibration measurement.
Multi-answer (controlled)	**MACE (Ours)**	✓	✓	✓	—

Single-correct multiple choice. MMLU-PRO (Wang et al., 2024) and ARC (Clark et al., 2018) present a small, closed option set per question with exactly one correct choice. While this format enables clean calibration measurement, the model's probability is confined to a predefined option set (e.g. A/B/C/D), entirely bypassing the open-ended probability dispersion over the full vocabulary that characterizes free-form generation. They fail **P2** and **P3**.

Multi-correct multiple choice. SATA-Bench (Xu et al., 2025) adopts a "select all that apply" format, permitting multiple correct options per question (**P1** ✓, **P2** ✓). Nevertheless, the model still operates over a closed, small option set rather than generating answers freely, limiting ecological validity for studying open-ended probability dispersion. More critically, the number of correct options per question is not systematically controlled but varies arbitrarily, making it impossible to trace calibration degradation as a function of answer cardinality. This fails **P3**.

Open single-answer QA. Benchmarks such as TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and Mintaka (Sen et al., 2022) target factoid questions with a single correct entity (e.g. *"Who wrote Hamlet?"* $\rightarrow$ *Shakespeare*). Because $|\mathcal{A}| = 1$ by construction, exhaustiveness is trivially satisfied (**P1** ✓), yet probability mass split cannot occur, failing both **P2** and **P3**. These benchmarks are therefore suitable for studying single-answer calibration but structurally incapable of diagnosing the multi-answer dispersion phenomenon.

Open multi-answer QA (non-exhaustive). WebQuestions (Berant et al., 2013) and ComplexWebQuestions (Talmor and Berant, 2018) derive answers from structured knowledge bases, naturally yielding some multi-entity answer sets (**P2** ✓). However, knowledge bases suffer from systematic incompleteness: entities and relations are frequently missing. As a result, the ground-truth answer sets are *not* exhaustive—a model may produce a factually correct answer that the knowledge base simply does not contain—violating **P1**. This incompleteness directly contaminates calibration evaluation, as correct predictions are miscounted as errors. Moreover, answer cardinality is an uncontrolled element of the structure, failing **P3**.

In summary, the landscape of existing benchmarks leaves a critical gap: *no current dataset simultaneously provides exhaustive ground truth, multi-answer questions, and controlled answer cardinality.* Closed-format benchmarks (single- and multi-correct MC) sidestep open-ended probability dispersion altogether; open single-answer benchmarks are structurally blind to mass split; and the only open multi-answer benchmarks rely on incomplete knowledge bases that corrupt calibration measurement. This gap makes it impossible to systematically study how probability mass split affects model calibration as the number of valid answers scales. To fill it, we introduce **MACE**, the first benchmark designed to satisfy **P1–P3** jointly, enabling controlled, fine-grained analysis of calibration under multi-answer probability dispersion.

B The MACE Benchmark

B.1 Domain Identification

Selecting appropriate factual domains is a prerequisite for a reliable and scalable benchmark. We begin by articulating three guiding principles:

1. **Answer Verifiability.** Every question must admit a *fixed-size* set of correct answers that can be verified objectively (i.e. non-subjectively). This rules out domains whose ground truth is opinion-dependent, time-sensitive with high update frequency, or otherwise ambiguous.

2. **Diversity under Verifiability.** Subject to the verifiability constraint, the chosen domains should collectively span heterogeneous knowledge categories so that the benchmark reflects a broad spectrum of real-world information needs rather than a single topical cluster.

3. **Large-Scale Data Availability.** Each domain must contain a sufficiently large pool of candidate questions (on the order of thousands) so that we can freely control the number of constraints per query and the overall evaluation cost, rather than being bottlenecked by scarce data.

LLM-assisted brainstorming. Guided by these principles, we conduct iterative brainstorming sessions with large language models (specifically GPT-4o) to enumerate candidate domains.

After multiple rounds of LLM proposals and manual expert deliberation, we converge on **six** representative domains that collectively satisfy all three criteria. Crucially, the ground-truth answers for every domain can be objectively verified against Wikipedia, ensuring reproducibility and low annotation cost:

- Honorable Award (Award): winners of well-documented prizes (e.g. Nobel, Academy Awards), whose complete recipient lists are maintained on Wikipedia with authoritative sourcing.

- Political Office (Office): holders of political offices with clear start/end dates, systematically catalogued in Wikipedia's officeholder tables.

- Regional Affiliation (Region): sovereign-state attributes such as organizational memberships and geographic properties, recorded in structured Wikipedia infoboxes.

- Mathematical Concept (Math): number-theoretic or combinatorial questions with provably unique answer sets, verifiable via Wikipedia's mathematical reference articles (e.g. lists of primes, Fibonacci numbers).

- Natural River (River): hydrological and geographic facts about rivers (e.g. length, tributaries, countries traversed), documented in Wikipedia with well-established reference data.

- Linguistic Culture (Language): official and national language designations, recorded in Wikipedia's country-level articles.

Limitations of LLM-assisted brainstorming. While LLM-driven enumeration accelerates the exploration of the domain space, it introduces several limitations that must be acknowledged:

(i) **Surface-level plausibility bias.** LLMs tend to propose domains that *sound* factual (e.g. "famous inventions," "influential scientists") without rigorously checking whether the answer sets are truly objective. Human verification is therefore indispensable.

(ii) **Popularity bias.** LLM suggestions are skewed toward domains that are well-represented in their training corpora (e.g. Western awards), potentially under-representing knowledge areas from less-documented cultures or disciplines.

(iii) **Hallucinated counter-examples.** When asked to provide domains that violate our criteria, LLMs occasionally fabricate plausible-sounding but incorrect justifications (e.g. claiming a domain is subjective when it is not). All counter-examples were therefore independently verified by the authors.

To mitigate these issues, every candidate domain proposed by the LLM underwent a two-stage human review: (1) a pilot feasibility check in which we attempted to construct at least 200 questions with verified ground truth, and (2) expert deliberation among the authors to confirm diversity and real-world relevance.

B.2 Triplet Collection

Following prior work on knowledge-grounded evaluation (Sen et al., 2022), we adopt Wikidata as our

Table 5: **Counts of subjects remaining after each filtering step. Raw**: Total count of retrieved subjects. **Pop.**: Count after popularity filter. **Val.**: Count after validity filter. **Man.**: Count after manual verification.

Domain	Raw	Pop.	Val.	Man.
Award	8,422	4,168	206	175
Office	7,306	5,000	289	159
Region	183	183	129	99
River	10,357	3,726	910	781
Language	7,251	3,830	1,012	847

primary knowledge source for entity retrieval, owing to its broad coverage, structured schema, and community-maintained quality.

We represent each piece of knowledge as a formal triplet $(s, r, \mathcal{O})$, where: (1) the **subject** s denotes the primary entity or anchor of the query (e.g., *Nobel Prize in Physics*); (2) the **relation** r denotes the semantic predicate connecting the subject to its answers (e.g., *laureate of*); (3) the **object set** $\mathcal{O} = \{o_1, o_2, \ldots, o_k\}$ contains all valid ground-truth entities satisfying relation r for subject s. This formalized triplet representation provides a unified, machine-readable structure that facilitates downstream processing, including automatic question generation, answer verification, and controlled cardinality manipulation. By varying the cardinality of the object set $|\mathcal{O}| \in \{1, 2, 4, 6\}$, we systematically control the difficulty and multi-answer complexity of each instance.

Except for the *Math* domain, which is generated through rule-based synthesis, all other domains are collected via the Wikidata SPARQL endpoint[2]. Specifically, for each domain, we query Wikidata to retrieve relevant subjects and their associated factual objects based on predefined relations (see Table 6). For instance, in the *Award* domain, we retrieve the entity *Nobel Prize in Physics* as the subject and obtain its *laureates* in a specific year as the objects, linked by the relation "is officially awarded to." To ensure alignment with common model knowledge, we restrict the temporal coverage to the years 1800–2023.

B.3 Heuristic-Based Filtering

To ensure data quality, we apply a two-stage filtering pipeline: a *Popularity Filter* followed by a *Validity Filter*. The *Popularity Filter* operates on the *subjects* of the knowledge triplets, while the *Validity Filter* is applied to both *subjects* and *objects*.

[2] https://query.wikidata.org/sparql

Popularity Filter. Raw Wikidata queries often return a long tail of obscure or niche entities—for example, minor regional awards with no public recognition, or little-known administrative subdivisions—that are unlikely to fall within the parametric knowledge of any language model. Including such entities would yield low-quality questions that models cannot reasonably be expected to answer, thereby introducing noise into the evaluation. To address this, we rank all entity labels within each domain by their 12-month aggregate Wikipedia pageview counts, which serve as a proxy for public familiarity. We apply two complementary retention criteria: (i) we retain any item whose pageview count exceeds 1,000, ensuring that all entities with a meaningful level of public recognition are preserved regardless of their intra-domain rank; (ii) within each domain, we further cap the candidate pool at the top 5,000 entries by pageview rank, keeping the dataset at a manageable scale for subsequent human annotation and manual quality review. Together, these criteria strike a balance between broad entity coverage and practical annotation cost, ensuring that the surviving subjects are both well-known enough to meaningfully probe model knowledge and tractable for rigorous human evaluation.

Validity Filter. Even after popularity-based pruning, entities retrieved from the Wikidata API frequently contain artifacts of incomplete or inconsistent curation. Common issues include missing human-readable labels (where only a raw QID such as `Q12345` is returned), empty or placeholder fields, broken outgoing URLs, and unresolved entity references that point to deleted or merged items. We therefore apply a *Validity Filter* that removes any triplet whose subject or object exhibits one or more of the following defects: (i) an absent or empty `rdfs:label` in the target language; (ii) a Wikidata identifier that cannot be resolved to a valid entity page; (iii) missing relational fields required by the domain schema (see Table 6).

Domain-Specific Refinement Criteria. Beyond these universal checks, we further impose domain-specific refinement criteria. A key motivation is that Wikidata's coverage is often *incomplete*: for a given subject, only a subset of its true objects may be recorded, which would introduce false negatives during evaluation—a model's correct answer could be wrongly penalized simply because it is absent from an incomplete ground-truth set. The domain-specific rules below therefore serve as heuristic

proxies for *completeness*: by retaining only entities whose Wikidata records exhibit signs of thorough curation (e.g., continuous yearly entries, sufficient object counts), we maximize the likelihood that the surviving triplets capture the full set of valid answers.

- **Award:** We retain only award subjects granted to a single laureate per year, with continuous yearly records and at least three objects per subject. The single-laureate-per-year constraint avoids partially recorded co-recipient lists, while requiring continuity and a minimum object count serves as evidence that the award's historical record on Wikidata is comprehensively maintained.
- **Office:** We preserve subjects held by exactly one person per year, requiring a minimum of ten recorded years per subject. Restricting to single-holder offices eliminates cases where Wikidata only logs a subset of concurrent holders, and the ten-year threshold filters out short-lived or sparsely documented positions whose records are likely incomplete.
- **Region:** We include only countries with at least three administrative subdivisions. Countries with fewer entries are often incompletely catalogued on Wikidata, and requiring a minimum count helps ensure the subdivision list is reasonably exhaustive.
- **River & Language:** We keep entities associated with no more than six countries to avoid overly generic mappings (e.g., English is spoken in dozens of countries, making exhaustive enumeration impractical and verification unreliable). Capping at six increases confidence that the Wikidata object set is complete. To control annotation cost, we further sample 600 instances from objects associated with a single subject.

B.4 Manual Verification

Although our heuristic filters substantially improve data quality, automated pipelines alone cannot guarantee factual correctness. We therefore conduct a rigorous multi-stage human verification process to validate all harvested triplets before inclusion in the final benchmark.

Independent Dual Annotation. Two annotators with domain expertise independently review each domain's entries along three dimensions: (i) *factual correctness*—whether the subject–relation–object mapping reflects a true real-world fact (e.g., verifying that a listed laureate did receive the specified award in the stated year); (ii) *completeness*—whether the object set $\mathcal{O}$ contains all valid answers rather than a partial subset; and (iii) *label consistency*—whether the surface-form labels faithfully correspond to their underlying Wikidata entities, ruling out disambiguation errors or stale redirects. Each triplet is cross-checked against its supporting Wikidata metadata (e.g., temporal qualifiers, geographic linkage, and official entity descriptions) to confirm semantic validity.

Adjudication and Agreement. Disagreements between the two annotators are resolved through a structured adjudication session: both annotators present their evidence, and a consensus label is produced only when both parties agree or a third senior annotator is consulted for tie-breaking. We measure inter-annotator reliability using Cohen's κ, obtaining $\kappa = 0.94$ averaged across all domains, indicating near-perfect agreement and confirming that the annotation guidelines are well-defined and consistently applied.

Iterative Spot-Check. After the initial annotation pass, we perform iterative spot-checking to catch residual errors. In each round, we draw a stratified random sample of 50 instances per domain—balanced across different object-set cardinalities—and revalidate them for factual soundness, formatting consistency, and correct cardinality assignment. Any identified errors trigger a targeted re-review of all entries sharing the same relation or subject category. This verification–revision loop is repeated until a full spot-check round surfaces zero errors; in practice, convergence was reached after two complete rounds.

B.5 QA Pair Generation

Design Rationale for Prompt Templates. All six domains share a unified **"Name one X that satisfies Y"** sentence frame (Table 7), where `X` specifies the entity type (e.g., person, country, number) and `Y` encodes one or more adjustable constraints (e.g., temporal range, geographic attribute, numeric interval). This deliberate design choice is motivated by our central evaluation goal: measuring how a model's confidence distributes across *successive* correct answers when the underlying question difficulty is held constant.

Table 6: **Detailed schema of relations and entity types across six factual domains.** Each row specifies the semantic category of subjects and objects, along with the predicate used to query the underlying knowledge base.

Domain	Subject Type	Relation ($s \rightarrow o$)	Object Type
AWARD	Award Name	*is officially awarded to*	Person
OFFICE	Political Position	*is officially held by*	Person
REGION	Sovereign Country	*contains as first-level division*	Administrative Division
MATH	Number Type	*includes as an instance*	Number
RIVER	River Name	*flows through geographically*	Country
LANGUAGE	Human Language	*is spoken widely in*	Country

Table 7: **Core prompt templates for each factual domain.** Parameterised slots (highlighted in `{braces}`) are dynamically filled to control the number of ground-truth answers k.

Domain	Prompt Template
AWARD	`Name one person who received` **`{award}`** `between` **`{lower_year}`** `and` **`{upper_year}`**.
OFFICE	`Name one person who officially assumed the office of` **`{office_name}`** `between` **`{lower_year}`** `and` **`{upper_year}`**.
REGION	`Name one first-level administrative division of` **`{region}`** `whose area lies between` **`{lower_area}`** `and` **`{upper_area}`** `km`^{2}.
MATH	`Name one` **`{number_type}`** `number between` **`{lower_num}`** `and` **`{upper_num}`**.
LANGUAGE	`Name one country where` **`{language_1}`**, **`{language_2}`**, `or` **`{language_3}`** `was predominantly spoken by native speakers.`
RIVER	`Name one country through which the` **`{river_1}`**, **`{river_2}`**, `or` **`{river_3}`** `flows.`

Concretely, the controlled template structure offers three methodological advantages:

- **Intra-domain Difficulty Homogeneity.** Because all instances within a domain differ only in the *parameterised slots* (e.g., `{award}`, `{lower_year}`–`{upper_year}`), surface-level confounds—such as varying syntactic complexity, question length, or pragmatic implicature—are effectively eliminated. This ensures that when a model's confidence decreases from the first correct answer to the k-th, the drop can be attributed to the model's internal knowledge recall dynamics rather than to heterogeneous question difficulty.
- **Unambiguous Answer Scope.** The "Name one ..." prefix explicitly signals that the expected output is a single entity per response turn. This removes ambiguity in automated answer matching and avoids partial-credit edge cases that would arise with open-ended phrasings such as "What are some ..." or "List ...".
- **Controlled Multi-answer Scaling.** The parameterised constraints (temporal windows, numeric intervals, disjunctive entity lists) act as *precision knobs*: widening a year range or adding an extra river to the disjunction increases the ground-truth answer count k without altering the question's syntactic form or cognitive demand. This mechanism is the foundation of our answer-count control ($k \in \{1, 2, 4, 6\}$).

We acknowledge that formulaic templates sacrifice naturalistic diversity. However, for a benchmark whose purpose is to isolate *confidence–rank dynamics under controlled difficulty*, eliminating surface variation is a deliberate and necessary methodological decision. We discuss extending the benchmark with more diverse question phrasings while preserving difficulty control in Section 7.

Retrieval-based Domains. Each domain consists of a unique set of factual subjects, where each subject is linked to one or more objects, forming foundational knowledge triplets $(s, r, \mathcal{O})$. To construct questions that elicit a specific number of correct answers ($k \in \{1, 2, 4, 6\}$), we dynamically adjust the parameterised constraints within each template via a *constraint relaxation* strategy, modulating the relational scope so that the resulting ground-truth set satisfies $|\mathcal{O}| = k$. This is realised through two complementary approaches:

1. **One-to-Many Expansion (Intra-subject)** — applied in *Award*, *Office*, and *Region*. For subjects that naturally possess multiple objects under a broad relation, we relax the constraint parameters while keeping the subject fixed. For example, in the *Award* domain, narrowing the year window `{lower_year}`–`{upper_year}` to a single year typically yields $k = 1$, whereas widening it to a five-year span can yield $k = 4$ or $k = 6$ verified laureates. Similarly, in *Region*, the area interval `{lower_area}`–`{upper_area}` is calibrated so that exactly k first-level administrative divisions fall within the specified range. In all cases, the relaxation is tuned to produce *exactly* k ground-truth objects.
2. **One-to-Few Aggregation (Cross-subject)** — applied in *Language* and *River*. When a single subject–relation pair cannot naturally yield k objects, we construct multi-answer instances by combining multiple subjects that share the same relational type into a disjunctive query. As shown in Table 7, the *Language* template includes up to three languages (`{language_1}`, `{language_2}`, `{language_3}`), and the *River* template includes up to three rivers. The number of disjuncts and the specific entities are selected so that the union of their associated countries totals exactly k. During aggregation, we verify that the merged object sets are *mutually exclusive* to prevent answer overlap. Since the combinatorial space of possible aggregations is vast, we cap the number of generated questions at 10,000 per domain.

Together, the two paths ensure that every domain can be populated across all four answer-count settings ($k \in \{1, 2, 4, 6\}$) while each question retains a well-defined scope and a uniform level of intrinsic difficulty.

Rule-based Domain. The *Math* domain is defined through five numeric category predicates (Table 8), each acting as a pseudo-label (e.g., prime numbers, perfect squares). Because these categories span broad numerical ranges, constructing questions around specific target numbers is unnecessary. Instead, we generate queries by uniformly sampling integers from the interval $[0,\ 1{,}000{,}000]$. For each sampled number n, the ground-truth label set $\mathcal{O}$ is computed deterministically by evaluating which of the five predicates n satisfies, and the numeric interval `{lower_num}`–`{upper_num}` in the template is set to a narrow window centred on n that contains exactly k qualifying numbers. This provides wide coverage of numerical reasoning behaviours while maintaining precise answer-count control. Similarly, given the virtually unlimited pool of candidate integers, we cap the total number of generated questions at 10,000.

Table 8: **Five mathematical number types** used to construct questions in the MATH domain. Each defines a distinct numeric sequence within $[0,\ 10^6]$.

Number Type	Definition
PRIME	$p > 1,\ p \in \mathbb{N},\ \nexists d \in (1, p) :\ d \mid p$
SQUARE	$n = k^2,\ k \in \mathbb{N}$
CUBE	$n = k^3,\ k \in \mathbb{N}$
FIBONACCI	$F_n = F_{n-1} + F_{n-2},\ F_1 = F_2 = 1$
TRIANGULAR	$T_n = \dfrac{n(n+1)}{2},\ n \in \mathbb{N}$

Quality Control. All generated QA instances undergo a multi-stage quality pipeline:

1. **Deduplication and Leakage Prevention.** We perform exact-match and near-duplicate removal (based on normalised question strings) and enforce that no answer entity appears in more than one domain, preventing cross-domain information leakage.
2. **Clustering-based Stratified Sampling.** To construct a representative and balanced dataset, we encode all candidate instances per domain-k partition with Sentence-BERT embeddings and apply K-Means clustering ($K = 500$). From each cluster we select the instance nearest to its centroid, yielding 500 instances per answer-count setting. Across four settings ($k \in \{1, 2, 4, 6\}$) this produces **2,000 distinct samples per domain** and a final dataset of **12,000 instances** over six domains.

C Experiment Setup

C.1 Calibration Methods

- **Prediction Entropy.**(Kadavath et al., 2022) To capture uncertainty across M generated responses, we compute the mean token-level entropy over all samples:

$$U_{\text{Prob Entropy}} = -\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{|r_j|} p_{z_i}\log p_{z_i}, \quad (3)$$

where p_{z_i} represents the normalized probability distribution over tokens within the j-th generation.

- **Length-Normalized Prediction Entropy.** (Malinin and Gales, 2020) To avoid bias toward longer sequences, we normalize the entropy of each sampled response by its length:

$$U_{\text{N-Prob Entropy}} = -\frac{1}{M}\sum_{j=1}^{M}\frac{1}{|r_j|}\sum_{i=1}^{|r_j|} p_{z_i}\log p_{z_i}. \tag{4}$$

This normalization ensures comparable uncertainty estimates across outputs of varying lengths.

- **Semantic Entropy.** (Kuhn et al., 2023) Given a set of semantic clusters $\{C_1, \ldots, C_K\}$ obtained from multiple sampled responses, we compute a probability distribution over clusters by aggregating the sentence-level probabilities. Specifically, the probability assigned to cluster C_k is

$$P(C_k) = \sum_{r_j \in C_k} P(r_j), \tag{5}$$

where $P(r_j)$ denotes the generation probability of response r_j. In standard semantic-entropy settings, these cluster probabilities are typically normalized to form a valid distribution:

$$\tilde{P}(C_k) = \frac{P(C_k)}{\sum_{i=1}^{K} P(C_i)}, \tag{6}$$

yielding $\tilde{P}(C_k) \in [0, 1]$ and $\sum_k \tilde{P}(C_k) = 1$.

The semantic entropy is then computed as the Shannon entropy of the normalized cluster distribution:

$$U_{\text{SE}} = -\sum_{k=1}^{K}\tilde{P}(C_k)\log\tilde{P}(C_k), \tag{7}$$

which quantifies the degree of semantic dispersion in the model's outputs—high values indicate diverse or inconsistent answer patterns, while low values correspond to concentrated semantic predictions.

From Entropy to Confidence. The three metrics above are measures of *uncertainty*: higher values indicate greater model hesitation. To convert them into a unified *confidence* score that is monotonically increasing with model certainty and bounded in $(0, 1]$, we apply an exponential transformation:

$$\text{Conf}(x) = e^{-H(x)}, \tag{8}$$

where $H(x) \in \{H_{\text{PE}}, H_{\text{NPE}}, H_{\text{SE}}\}$ denotes the entropy value for a given instance x. When entropy is zero (the model is maximally certain), the confidence equals 1; as entropy grows, confidence decays smoothly toward 0. This transformation preserves the relative ordering of instances and enables direct comparison with other confidence estimators on a common scale throughout our analysis.

- **Verb.** (Jiang et al., 2021) This method elicits confidence by prompting the model to verbalize a numeric confidence score alongside its answer. Given a question x, the model generates a textual response containing both the answer $\hat{y}$ and a confidence expression. The confidence value C is then extracted from the generated text:

$$C_{\text{Verb}} = \text{LLM}(x). \tag{9}$$

- **Verb$_{\text{Topk}}$.** (Xiong et al., 2023) This variant extends the verbalization setting by asking the model to provide k distinct candidate answers $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_k\}$ along with their respective confidences $\{C_1, C_2, \ldots, C_k\}$. The final confidence is defined as the maximum confidence among semantically valid candidates:

$$C_{\text{Verb}_{\text{Topk}}} = \max_{1\le i\le k}\left(C_i \cdot \mathbb{I}\{\hat{y}_i \in \mathcal{Y}^*\}\right), \tag{10}$$

where $\mathcal{Y}^*$ denotes the set of semantically valid answers.

- **Consistency.** (Manakul et al., 2023) Consistency score measures the proportion of model samples that agree with the aggregated prediction $\tilde{Y}$ among M sampled outputs:

$$C_{\text{consistency}} = \frac{1}{M}\sum_{i=1}^{M}\mathbb{I}\{\hat{Y}_i = \tilde{Y}\}, \tag{11}$$

- **Weighted Consistency**(Xiong et al., 2023) To incorporate the self-reported confidence C_i of each sampled answer, we further introduce a weighted variant:

$$C_{\text{conf}} = \frac{\sum_{i=1}^{M}\mathbb{I}\{\hat{Y}_i = \tilde{Y}\}\times C_i}{\sum_{i=1}^{M} C_i}, \tag{12}$$

where higher-confidence answers contribute proportionally more to the final score. When C_i is obtained from the *Verb* or *Verb$_{Topk}$* methods, this variant corresponds to *Consistency-Verb* and *Consistency-Verb-Topk*, respectively.

- **Perplexity.**(Manakul et al., 2023) Perplexity is computed as the average negative log-likelihood of token probabilities:

$$U_{\text{Perplexity}} = -\frac{1}{|r|}\sum_{i=1}^{N} \log p_{z_i}, \qquad (13)$$

where r denotes the generated sequence and p_{z_i} is the model-assigned probability for the i-th token.

- **P$_{\text{True}}$-Consis/P$_{\text{True}}$-Prob** (Kadavath et al., 2022). These methods estimate model confidence by prompting the LLM to explicitly verify whether its previously generated answer is correct. The verification prompt x' asks the model to respond with "`True`" or "`False`", and the resulting confidence is defined as the probability of the token "`True`":

$$C_{\text{P}_{\text{True}}} = p(z_{\text{true}} \mid x'). \qquad (14)$$

We implement two variants: (1) In the **P$_{\text{True}}$-Consis** variant, confidence is obtained by performing multiple sampling runs and taking the empirical frequency of the "`True`" response across samples; (2) In the **P$_{\text{True}}$-Prob** variant, we directly use the model's internal probability (logit-derived softmax value) assigned to the "`True`" token, without repeated sampling.

- **P$_{\text{True}}$-Consis-Cand/P$_{\text{True}}$-Prob-Cand** (Kadavath et al., 2022). To better handle multi-answer scenarios, we extend the verification prompt x' by including a set of candidate answers $\{a_1, a_2, \ldots, a_K\}$, allowing the model to evaluate its response in the context of multiple plausible answers:

$$C_{\text{P}_{\text{True}}\text{-Consis-Cand}} = p(z_{\text{true}} \mid x', \{a_1, a_2, \ldots, a_K\}). \qquad (15)$$

Similarly, two variants are considered: (1) In the **P$_{\text{True}}$-Consis-Cand** variant, confidence is computed as the empirical frequency of "`True`" responses across multiple sampling runs; (2) In the **P$_{\text{True}}$-Prob-Cand** variant, the probability assigned to the "`True`" token from the model's logits is used directly as the confidence score.

- **Self$_{\text{Ask}}$.** (Tian et al., 2023) This method prompts the model to *self-assess* and directly verbalize its own confidence score after producing an answer. Given a question–answer pair $(x, \hat{y})$, the model is re-prompted with a meta-instruction asking it to output a numerical confidence value (e.g., "0.85") that reflects how certain it is about $\hat{y}$. The resulting confidence is defined as:

$$C_{\text{Self}_{\text{Ask}}} = \text{LLM}(x', \hat{y}), \qquad (16)$$

where x' denotes the self-assessment prompt and LLM$(\cdot)$ returns a scalar confidence value verbalized by the model.

- **Self$_{\text{Ask}}$-Cand.** (Tian et al., 2023) To better account for uncertainty among multiple plausible answers, this variant augments the verification prompt with a set of candidate answers $\{a_1, a_2, \ldots, a_K\}$. The model is instructed to assess its confidence in $\hat{y}$ given this broader context. The confidence is computed as:

$$C_{\text{Self}_{\text{Ask}}\text{-Cand}} = \text{LLM}(x', \hat{y}, \{a_1, a_2, \ldots, a_K\}), \qquad (17)$$

where x' includes both the original answer and the candidate list as context for self-evaluation.

C.2 Experiment Details

Correctness Evaluation. Given the diversity of output formats, we use `GPT-4o` as our primary evaluator, following (Liu et al., 2023). An answer is deemed correct if GPT-4o determines that it matches any of the ground-truth answers. GPT-4o is also employed to assess semantic consistency between pairs of responses. To validate the reliability of these automatic judgments, we perform human evaluation on 300 randomly sampled instances, observing a Cohen's κ of 0.92, which indicates high agreement among annotators.

Inference Framework. All experiments are conducted using **vLLM** (Kwon et al., 2023) as the inference backend, which provides efficient batched decoding and native access to token-level log-probabilities required by probability-based confidence estimators.

Sampling Configuration. We adopt two distinct sampling regimes depending on the confidence estimation method:

Figure 8: Domain-specific accuracy and confidence under different ground-truth sizes.

- **Single-sample methods** (verbalized confidence, post-hoc elicitation, self-verification, token probability, and accuracy evaluation): temperature is set to $T = 0$ (greedy decoding) to obtain deterministic outputs.
- **Multi-sample methods** (consistency-based and semantic-entropy-based estimators): temperature is set to $T = 1$ and we draw $M = 20$ independent samples per question to capture the diversity of the model's output distribution.

Entropy-to-Confidence Conversion. For all entropy-based estimators (Prediction Entropy, Length-Normalised Prediction Entropy, and Semantic Entropy), the raw entropy value H is converted into a confidence score via the exponential transformation

$$\text{Conf} = e^{-H},$$

as described in Eq. (8). This ensures that all methods produce scores on a common $(0, 1]$ scale, enabling direct cross-method comparison.

LLM-based Evaluation. We use `GPT-4o` as the automated evaluator for two critical components of our pipeline:

1. **Answer Correctness Judgment.** Exact-match and rule-based heuristics perform poorly on our benchmark due to surface-form variations (e.g., "U.S.A." vs. "United States", partial name matches, and aliasing across languages). We therefore prompt `GPT-4o` with the question, the model's response, and the ground-truth answer set, asking it to return a binary `True`/`False` verdict.
2. **Semantic Similarity for Multi-sample Methods.** For consistency-based and semantic-entropy-based estimators, pairwise semantic equivalence between sampled responses must be determined to form semantic clusters. We again rely on `GPT-4o` to judge whether two responses convey the same factual answer, as embedding-based similarity alone cannot reliably distinguish near-miss paraphrases from genuinely distinct answers.

To validate the reliability of these automatic judgments, we perform human evaluation on 300 randomly sampled instances (100 from answer correctness, 200 from pairwise similarity), observing a Cohen's κ of **0.92**, indicating near-perfect agreement between `GPT-4o` and human annotators.

Aggregation Strategy. All reported metrics follow a **macro-averaging** scheme across the six domains to prevent any single domain from dominating the aggregate. The specific aggregation proce-

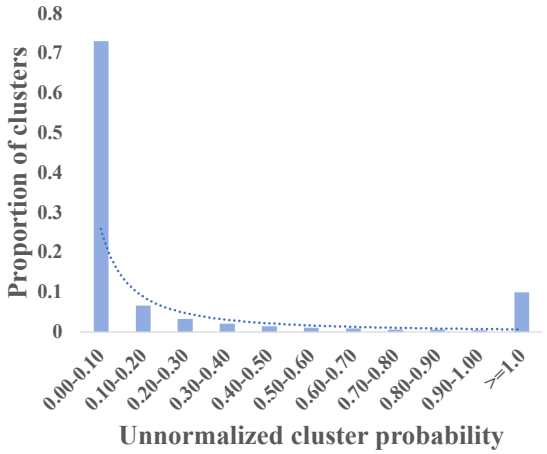

Figure 9: **Distribution of unnormalised cluster probabilities** $P_{\text{raw}}(C_m) = \sum_{r_j \in C_m} P(r_j)$ for **LLaMA-3.1-70B-Instruct**. Each bar shows the fraction of semantic clusters whose cumulative probability falls within the corresponding bin. Since probabilities are summed over $M = 20$ samples without renormalisation, values can exceed 1. Over 70% of clusters concentrate in $[0,\ 0.1)$, motivating our filtering threshold $\tau = 0.1$ (see §D.3). The dotted curve is a kernel density estimate for visual reference.

dure differs by metric type:

1. **Accuracy and Confidence.** For each answer-count setting ($k \in \{1, 2, 4, 6\}$), we compute the metric independently within each domain and then report the unweighted mean across all six domains. For the MATH domain, which is further subdivided into five number-type labels, we first average across labels before contributing a single domain-level score, ensuring that MATH carries the same weight as any other domain.
2. **AUROC.** Because AUROC requires a sufficient mixture of positive and negative instances, we first *pool* all instances within a domain across the relevant answer-count settings—either a single k when reporting per-setting AUROC, or all four settings ($k \in \{1, 2, 4, 6\}$) when reporting the mixed-k AUROC—and compute one AUROC value per domain on this pooled set. The final reported number is then the unweighted mean of the six domain-level AUROC scores.
3. **Inter-domain averaging.** In both cases above, the final aggregate is an *unweighted* mean over the six domain-level values. This macro-average treats every domain as an equally important evaluation scenario, preventing high-accuracy domains from inflating the overall metric (see §C.3 for further discussion).

C.3 Domain Effects: Relation Between Question Type and Difficulty

We analyze the domain-level calibration behavior of **LLaMA-3.1-70B-Instruct** across varying ground-truth set sizes. As illustrated in Figure 8, all configurations (`1a/2a/4a/6a`) exhibit a highly consistent trend: domains characterised by higher accuracy (lower intrinsic difficulty) also demonstrate higher confidence levels.

Why Macro-Average as the Primary Metric. The six domains span a wide difficulty spectrum—from near-ceiling accuracy in *Language* and *River* to substantially lower accuracy in *Region* and *Award*. If we were to report a *micro-average* (i.e., pooling all instances regardless of domain), the aggregate metric would be dominated by the high-accuracy, low-difficulty domains, masking calibration deficiencies in harder domains where reliable confidence estimation matters most. To prevent this skew, we adopt the **macro-average**—computing each metric independently within every domain and then averaging across domains—as our primary reporting metric throughout the paper. This treats every domain as an equally weighted evaluation scenario and yields a more faithful picture of global calibration quality, especially for safety-critical applications where a model's worst-case domain performance is at least as important as its best-case performance.

Structural Origins of Difficulty Variation. Crucially, the difficulty gap across domains is *not* an artefact of inconsistent question phrasing—all six domains share the same "Name one X that satisfies Y" template (Table 7; Appendix §B.5). Instead, it originates from the two fundamentally different *construction paths* used to populate multi-answer instances:

- **Cross-subject Aggregation** (*Language*, *River*). Multiple subjects are composed into a single disjunctive query (e.g., "Name one country where language A, B, or C is spoken"). Each disjunct maps to a small, well-separated set of countries, so the model only needs to recall *any one* member from a union of compact, high-frequency fact sets. The resulting answer space is narrow and semantically unambiguous, yielding the **lowest difficulty**.
- **Intra-subject Expansion** (*Award*, *Office*, *Region*). A single subject's constraint window is

widened (e.g., a broader year range for awards, a larger area interval for administrative divisions) to harvest k ground-truth objects. Although each question still targets one coherent topic, the expanded window opens up a *large candidate pool* of entities that share similar contextual attributes (e.g., overlapping tenures, comparable geographic areas). The model must perform fine-grained attribute linking—matching a specific temporal or spatial constraint to the correct entity among many plausible alternatives—which substantially increases **retrieval complexity and semantic confusability**.

The *Math* domain occupies an intermediate position: it bypasses factual retrieval entirely and instead requires recognition of abstract numeric properties (primality, perfect squares, etc.) over uniformly sampled integers. This is cognitively simpler than the attribute-linking in Intra-subject domains, yet harder than the disjunctive enumeration in Cross-subject domains, because the model must apply rule-based reasoning rather than surface-level association.

In summary, the difficulty hierarchy is a direct consequence of the construction methodology described in Appendix §B.5. Because difficulty is structurally determined by the domain's construction path rather than by surface-level question variation, it remains stable across all answer-count settings ($k \in \{1, 2, 4, 6\}$).

D Results and Analysis

D.1 Consistent Trends Across Models and Scales.

We report the **confidence** of all evaluated models under different numbers of valid answers (1a/2a/4a/6a), as shown in Tables 17–24. Note that for API-based models (e.g., GPT-4o, GPT-4o-mini, and DeepSeek-V3), we do not report methods that rely on logit probabilities or internal token distributions, such as Prob-Entropy, N-Prob-Entropy and Sem Entropy. For entropy-based methods (Prob Entropy, N-Prob Entropy, and Sem Entropy), we convert entropy H into a confidence score via $\text{conf} = e^{-H}$ to ensure a consistent and comparable scale with other methods, where higher values uniformly indicate greater confidence.

Across model families and parameter scales, a clear and consistent pattern emerges: model accuracy follows a steady upward trend as the number of valid answers increases. However, the model confidence exhibits a diametrically opposite decline over the same range. This diverging behavior reveals a significant misalignment between the models' actual performance and their certainty: as the task environment becomes more "favorable" (i.e., more valid answers are available), the models paradoxically become increasingly "underconfident." Such a monotonic decline in confidence despite rising accuracy demonstrates that existing calibration methods are unable to remain stable when confronted with multi-answer variation, highlighting a fundamental limitation of current approaches.

D.2 Why Does Confidence Decline? A Per-Method Analysis

The consistent confidence decline observed across all method families (§D.1) naturally raises the question: *what is the underlying mechanism for each category of methods?* We analyze the three major families separately below.

Consistency-based methods. Methods such as Consistency and Semantic Entropy estimate confidence by measuring agreement among multiple sampled responses. Through case-level inspection, we find that the confidence drop is directly attributable to **answer diversity fragmenting the generation probability mass**. When a question has a single valid answer, the model's sampled responses tend to converge on one surface form, yielding high inter-sample agreement. As the number of valid answers grows, the model distributes its generations across multiple equally correct but textually distinct answers. For instance, when asked *"Name one person who received the Nobel Prize in Physics in 1995,"* the model alternates among "Martin Lewis Perl" and "Frederick Reines"—both correct—across samples. Consistency-based methods interpret this diversity as disagreement, producing a low confidence score that is indistinguishable from genuine uncertainty caused by alternation between correct and incorrect answers.

Probability-based methods. For generation-probability methods (e.g., Perplexity), the confidence decline might appear surprising at first, since these methods operate on a single greedy or sampled response rather than on multi-sample agreement. To investigate the root cause, we measure the entropy of the **first generated token's top-20 logits** for Perplexity on LLaMA-3.1-70B-Instruct, stratified by answer count:

	1a	2a	4a	6a
First-token entropy	1.22	1.30	1.53	1.68

Table 9: Average entropy over the top-20 logits of the first generated token under different answer counts (LLaMA-3.1-70B-Instruct).

The monotonic increase in first-token entropy confirms that, even at the very onset of generation, the model's probability mass is more dispersed when more valid answers exist. Intuitively, the model "hesitates" among multiple plausible starting tokens (e.g., "Martin" vs. "Frederick"), each leading to a different correct answer. This initial hesitation propagates through the entire sequence, resulting in lower overall sequence probability and thus lower Perplexity-based confidence. In other words, the same answer-diversity mechanism that afflicts consistency-based methods also affects probability-based methods, but manifests at the *token distribution* level rather than the *sample agreement* level.

Double-turn methods. In contrast to the above two families, double-turn methods (e.g., P_{True}, Self-Ask) exhibit a noticeably smaller confidence decline across answer-count settings. As shown in Table 2, P_{True}-Consis-Cand maintains confidence scores of 88.1, 87.9, 87.1, and 87.9 for 1a through 6a, respectively—a variation of less than 1.2 points. This robustness stems from the fact that double-turn methods **decouple confidence estimation from the generation process**. Rather than deriving confidence from how the answer was produced (token probabilities or sampling agreement), these methods pose a separate verification query—e.g., *"Is the following answer correct: Martin Lewis Perl? (True/False)"*—and assess confidence based on the model's *judgment* ability. Since verifying a single correct answer does not require the model to choose among alternatives, the verification step is largely unaffected by how many other valid answers exist. However, this advantage comes at the cost of an additional inference pass per question, and as shown in our main results (Tables 3–16), double-turn methods do not consistently outperform our proposed SCA in AUROC, which achieves competitive or superior calibration while remaining a single-turn approach.

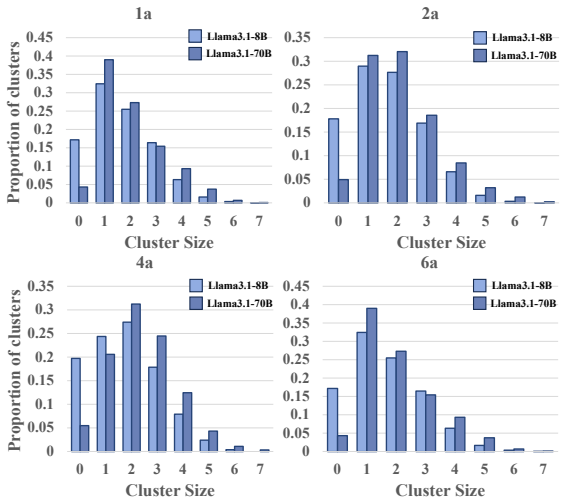

Figure 10: **Distribution of cluster sizes of llama family.**

D.3 Analysis of Cluster Probability Distribution

We apply a token-level probability-based filter and retain only clusters whose cumulative probability exceeds a threshold of $\tau = 0.1$.

Why Unnormalised Cluster Probabilities. A key design decision in our semantic-entropy pipeline is to measure each cluster's importance by its **unnormalised cumulative probability**, defined as the sum of token-level generation probabilities across all member responses:

$$P_{\text{raw}}(C_m) \;=\; \sum_{r_j \in C_m} P(r_j),$$

where $P(r_j)$ is the sequence-level probability of the j-th sampled response. Because we draw $M = 20$ samples per question, the raw sum $P_{\text{raw}}(C_m)$ is *not* constrained to $[0, 1]$—clusters that capture the model's dominant output mode can accumulate values exceeding 1.

We deliberately avoid renormalising these probabilities across clusters (i.e., converting them into a simplex distribution $\hat{P}(C_m) = P_{\text{raw}}(C_m)/\sum_m P_{\text{raw}}(C_m)$). The reason is that normalisation would *inflate* the relative weight of low-probability clusters: even when *all* clusters carry negligible absolute mass (e.g., the model is highly uncertain and scatters probability across many fragmented outputs), normalisation would still assign some cluster a large *relative* share, creating the false impression of a confident semantic mode. By operating on unnormalised probabilities, we preserve the absolute scale of the model's output dis-

tribution, so that a cluster's importance directly reflects the model's genuine probability commitment to that semantic content.

Empirical Distribution and Threshold Selection. Figure 9 shows the distribution of $P_{\text{raw}}(C_m)$ across all clusters produced by **LLaMA-3.1-70B-Instruct**. The distribution exhibits a striking **bimodal** structure:

- A dominant spike in the $[0,\ 0.1)$ bin, accounting for over **70%** of all clusters. These are low-mass clusters that each capture only a tiny fraction of the 20 samples—typically one or two responses with individually low generation probabilities. Manual inspection confirms that they predominantly consist of fragmented, degenerate, or semantically irrelevant outputs.
- A secondary concentration at $P_{\text{raw}} \geq 1.0$ (approximately **10%** of clusters), representing the model's dominant answer modes where multiple high-probability samples converge on the same semantic content.

Between these two modes, the bins from $[0.1,\ 1.0)$ are sparsely populated, forming a clear **probability gap** that provides a natural decision boundary. We therefore set the filtering threshold at

$$\tau \ = \ 0.1,$$

retaining only clusters with $P_{\text{raw}}(C_m) \geq 0.1$. The rationale for this specific value is threefold:

1. **Empirical gap alignment.** The threshold sits precisely at the boundary between the dominant near-zero spike and the sparsely populated mid-range bins. Any value in the interval $[0.05,\ 0.15]$ would produce similar filtering outcomes; 0.1 is chosen as the round value at the centre of this stable region.
2. **Intuitive interpretation under 20 samples.** With $M = 20$ draws, $P_{\text{raw}}(C_m) = 0.1$ corresponds roughly to a cluster that either (a) contains a single sample with generation probability ≈ 0.1, or (b) aggregates two samples each with probability ≈ 0.05. In both cases the cluster represents less than 0.5% of the total probability budget ($\sum_m P_{\text{raw}}(C_m)$), making it negligible for uncertainty estimation.
3. **Cross-domain and cross-model stability.** We verified that the $[0,\ 0.1)$ spike persists across all six domains and all evaluated models, confirming that $\tau = 0.1$ is not over-fitted to a single experimental condition.

Table 10: **Spearman correlation of confidence with knowledge coverage** across ground-truth(GT) set sizes using *consistency* method.

Model	**1a**	**2a**	**4a**	**6a**
Qwen2.5-7B	0.009	-0.000	-0.000	-0.012
Qwen2.5-14B	-0.443	-0.438	-0.427	-0.421
Qwen2.5-32B	-0.421	-0.421	-0.407	-0.399
Qwen2.5-72B	-0.616	-0.551	-0.533	-0.516
Llama-3.1-8B	0.242	0.227	0.325	0.366
Llama-3.1-70B	-0.393	-0.314	-0.167	-0.055
GPT-4o-mini	-0.590	-0.563	-0.567	-0.544
GPT-4o	-0.821	-0.779	-0.729	-0.684
DeepSeek-V3	-0.854	-0.827	-0.820	-0.807

Applying this threshold yields a compact set of *effective* semantic clusters that faithfully represent the model's meaningful output modes, filtering out sampling noise and providing a cleaner basis for computing semantic entropy and analysing confidence–rank dynamics.

D.4 Spearman metric cross Knowledge Coverage

Validity of Spearman Correlation. Spearman's rank correlation is sensitive to tied ranks and can produce misleading results when the distribution of one variable is heavily concentrated at a single value. We therefore verify that the effective answer-space size (i.e., the number of semantic clusters retained after filtering) is sufficiently dispersed to support a meaningful rank-based analysis.

Figure 10 shows the distribution of cluster sizes across both models and all four ground-truth settings (`1a/2a/4a/6a`) after applying the probability threshold $\tau = 0.1$. Two key observations confirm the validity of Spearman correlation in our setting:

1. **No single cluster size dominates.** In every setting, cluster sizes span a range from 0 to 6+, and no individual bin accounts for more than 40% of the total mass. This rules out the degenerate case where most instances collapse to the same cluster count—which would produce excessive tied ranks and inflate or deflate the correlation coefficient.
2. **Systematic shift with increasing ground-truth cardinality.** As the number of ground-truth answers grows from $k = 1$ to $k = 6$, the cluster-size distribution shifts smoothly toward larger values without collapsing into a narrow mode or generating extreme outliers. This gradual expansion confirms that the effective answer space grows in a stable manner, providing

a well-graded ordinal variable that Spearman correlation can reliably capture.

Correlation Results. Table 10 reports the Spearman correlation between model confidence and knowledge coverage using *consistency* method. Across multiple models, we observe a consistent **negative correlation**: as the effective answer space expands (i.e., more semantically distinct clusters survive filtering), the model's reported confidence *decreases*. This finding indicates that higher confidence does not reflect broader factual coverage; rather, models tend to be most confident when their output distribution concentrates on a small number of semantic modes, even though a wider set of valid answers exists.

Notably, **larger models exhibit a stronger negative correlation** than their smaller counterparts. This suggests that as model capacity increases, the confidence score becomes *more* sensitive to the expansion of the effective answer space—precisely the regime where miscalibration is most consequential. This observation is consistent with our earlier findings (Section 5), where larger models display steeper confidence degradation as the number of ground-truth answers grows, reinforcing the conclusion that scale alone does not mitigate—and may even exacerbate—the confidence–coverage misalignment under multi-answer settings.

E Semantic Confidence Aggregation

E.1 Generalization Across Models

To evaluate whether our method generalises beyond the primary experimental models, we extend the full evaluation suite to additional architectures and scales: **Qwen-2.5-7B-Instruct** (Table 12), **Qwen-2.5-14B-Instruct** (Table 13), **Qwen-2.5-32B-Instruct** (Table 14), **Qwen-2.5-72B-Instruct** (Table 15), and **LLaMA-3.1-8B-Instruct** (Table 16).

Across the majority of models and settings, SCA achieves the best or second-best AUROC. On the smaller-scale models (7B and 14B) and LLaMA-8B, SCA dominates all baselines under every answer-mixture configuration, and its advantage becomes more pronounced as the answer mixture broadens (i.e., moving from **[1]** to **[1,2,4,6]**), confirming that the method's strength lies precisely in the multi-answer regime where existing estimators degrade most severely.

On the 72B Qwen and 70B LLaMA models, SCA again achieves the best or near-best AUROC in most settings, demonstrating that its effectiveness is not limited to smaller-scale models.

Furthermore, SCA with $\tau = 0$ (no filtering) performs on par with or better than the threshold-tuned variant in nearly all cases across all models, suggesting that the method is robust to the choice of filtering threshold and does not require per-model hyper-parameter tuning.

Statistical Significance. To rigorously assess whether SCA's improvements are statistically meaningful, we conduct a bootstrap significance test. For each answer-mixture setting, we randomly sample 1,000 instances with replacement from the corresponding test pool and compute the AUROC for both SCA and every baseline method. This resampling procedure is repeated 1,000 times, yielding an empirical distribution of AUROC differences. A baseline is marked with † in Tables 12–16 if the two-sided p-value exceeds 0.05 (i.e., the difference from SCA is *not* statistically significant). Across all models and settings, the vast majority of baselines show $p < 0.05$, confirming that SCA's advantage is statistically significant in the overwhelming majority of comparisons. The few non-significant cases (marked †) occur mostly in the single-answer setting **[1]**, where the multi-answer challenge is absent and several strong baselines already perform near-optimally.

These results, combined with the primary findings on LLaMA-3.1-70B-Instruct (Table 3), demonstrate that SCA is **model-agnostic** and **scale-agnostic**, maintaining strong performance across different model families (LLaMA, Qwen) and parameter scales (7B–72B). While SCA is the clear overall winner as a *single-turn* calibration method, the 32B results highlight that sufficiently capable models may benefit from double-turn probing when the additional inference cost is acceptable—an interesting direction for future work on adaptive confidence estimation.

E.2 Results on NQ Dataset

To further assess the generalization of SCA on single-answer datasets, we randomly sample 500 questions from NQ-open (Kwiatkowski et al., 2019) and evaluate calibration performance (AUROC). The results are shown in Table 11.

SCA ($\tau = 0$) achieves 75.6 AUROC without any threshold tuning, outperforming all double-turn methods and confidence aggregation baselines. With an optimal threshold ($\tau = 0.9$), SCA matches

Table 11: **AUROC score on NQ-open** (500 randomly sampled questions) of different calibration methods on LLaMA-3.1-70B-Instruct. $\tau = 0$ denotes no-filtering, while $\tau \neq 0$ uses the threshold value tuned on the development set. **Bold** marks the best, underlined the second-best.

Method	NQ
Single-turn	
Question-level	
Prob Entropy	75.9
N-Prob Entropy	76.2
Sem Entropy	76.0
Answer-level	
Verb	64.0
Verb-Topk	69.5
Consis	**78.0**
Consis-Verb	73.3
Consis-Verb-Topk	74.6
Perplexity	75.7
Double-turn	
P_{True}-Consis	65.1
P_{True}-Prob	70.1
P_{True}-Consis-Cand	59.0
P_{True}-Prob-Cand	72.4
Self-Ask	67.4
Self-Ask-Cand	67.8
Confidence Aggregation Baselines	
SNCA ($\tau = 0.15$)	74.8
SNCA ($\tau = 0$)	50.0
SFCA ($\tau = 0.35$)	73.7
SFCA ($\tau = 0$)	50.0
Ours	
SCA ($\tau = 0.9$)	76.2
SCA ($\tau = 0$)	75.6

the best single-turn question-level method (N-Prob Entropy, 76.2), demonstrating strong generalization on single-answer benchmarks.

Table 12: **AUROC score on Qwen-2.5-7B-Instruct** of different calibration methods under increasing answer mixture. $\tau = 0$ denotes no-filtering, while $\tau \neq 0$ uses the threshold value tuned on the development set. **Bold** marks the best, underlined the second-best in each column. † indicates the difference from our SCA method is not statistically significant ($p > 0.05$).

Method	[1]	[1,2]	[1,2,4]	[1,2,4,6]
Single-turn				
Question-level				
Prob Entropy	77.7	77.5	75.7	74.4
N-Prob Entropy	77.5†	76.1	74.3	73.6
Sem Entropy	75.6	75.3	73.4	72.1
Answer-level				
Verb	58.3	57.0	56.4	55.2
Verb-Topk	61.7	56.9	52.8	51.8
Consis	78.4	78.0	76.3	75.3
Consis-Verb	75.5	74.7	74.3	73.7
Consis-Verb-Topk	75.6	74.2	73.6	72.4
Perplexity	78.8†	77.9	76.0	75.4
Double-turn				
P_{True}-Consis	71.3	67.3	67.8	67.3
P_{True}-Prob	76.2†	71.7	72.2	71.0
P_{True}-Consis-Cand	69.0	69.3	68.0	67.0
P_{True}-Prob-Cand	77.6†	76.6	74.6	73.3
Self-Ask	69.2	65.9	64.9	62.8
Self-Ask-Cand	76.5	75.0	72.7	71.3
Confidence Aggregation Baselines				
SNCA ($\tau = 0.2$)	74.1	74.5	72.7	71.6
SNCA ($\tau = 0$)	50.0	50.0	50.0	50.0
SFCA ($\tau = 0.15$)	76.4	76.0	75.5	74.3
SFCA ($\tau = 0$)	50.0	50.0	50.0	50.0
Ours				
SCA ($\tau = 0.1$)	79.9	79.7	78.1	77.0
SCA ($\tau = 0$)	**80.2**	**79.8**	**78.2**	**77.1**

Prompt for LLM-Assisted Domain Brainstorming

System Prompt:

You are an expert research assistant helping to design a factual-QA benchmark. Your task is to brainstorm knowledge domains that satisfy a set of strict selection criteria provided by the user. You should propose candidate domains, justify each one, and also provide counter-examples of domains that appear suitable but actually fail the criteria.

User Prompt:

`## Instruction`

I am constructing a factual-QA benchmark. Each question asks the model to produce a *complete set* of entities that satisfy several constraints simultaneously. I need to choose knowledge domains that satisfy **all** of the following criteria:

1. **Objective & fixed-size answers.** The ground-truth answer set is uniquely determined and easily verifiable against public sources (e.g. Wikipedia, official records). The number of correct answers is finite and well-defined.
2. **Diversity.** The domains should span different areas of human knowledge (science, geography, politics, culture, etc.).
3. **Scalability.** Each domain should support thousands of questions with varied constraint combinations, so that the benchmark can be scaled up cheaply.

Please suggest 15–20 candidate domains with a brief justification for each. **Also provide counter-examples**—domains that *seem* suitable but actually violate one or more criteria—with an explanation of why they fail.

`## Counter-Example Guidance`

Below are several examples we provide to guide your reasoning. Each illustrates a domain that initially appears suitable but actually violates one or more of the above criteria:

• "Best-selling novels of a given decade"—answer sets are publisher-dependent and vary across ranking sources, violating **Criterion 1** (verifiability).

• "Famous paintings in a particular art movement"—the membership of an art movement is subjective; experts disagree on boundary cases, violating **Criterion 1**.

• "Endangered species in a region"—the IUCN Red List is updated continuously, so the answer set is temporally unstable, making large-scale verification expensive and violating **Criterion 1 & 3**.

• "Chemical elements with a specific property"—while verifiable, the total number of elements is only 118, severely limiting constraint diversity and violating **Criterion 3**.

• "Olympic host cities"—verifiable and objective, but the total pool is small (<40 entries), making it hard to construct thousands of distinct questions, violating **Criterion 3**.

Based on the criteria and counter-examples above, propose your candidate domains.

Prompt Template for Verb

System Prompt:

You are a knowledgeable assistant. When answering factual questions, you must provide your answer along with a numeric confidence score that reflects your genuine certainty about the answer being correct.

User Prompt:

`## Instruction`

Please answer the following question and provide your confidence as a numeric value between 0 and 1 (e.g., 0.85 means you are 85% confident). Use two-decimal precision for your confidence value.

`## Input`

Question: {question}

```
## Output Format

Answer: {answer}
Confidence:
{confidence between 0 and 1
two-decimal precision}
```

Prompt Template for $\text{Verb}_{\text{Topk}}$

System Prompt:

You are a knowledgeable assistant. When answering factual questions, you must provide your top-k most likely answers along with a numeric confidence score for each. Your confidence should reflect your genuine certainty about each answer being correct.

User Prompt:

`## Instruction`

Please list your top-k most likely answers to the following question. For each answer, report your numeric confidence between 0 and 1 (e.g., 0.85 means 85% confident). Use two-decimal precision for all confidence values.

`## Input`

Question: {question}

`## Output Format`

```
1. Answer: {answer_1},
Confidence: {confidence_1}
2. Answer: {answer_2},
Confidence: {confidence_2}
... up to k answers.
```

Prompt Template for P_{True}

System Prompt:

You are a rigorous fact-checking assistant. Given a question and a previously generated answer, you must judge whether the answer is factually correct. Respond strictly with either `True` or `False`—no explanations, no additional text.

User Prompt:

`## Instruction`

You will be given a question and the model's previous answer. Please respond with `True` if the answer is factually correct, or `False` if it is incorrect. Answer strictly with either "True" or "False" only.

`## Input`

Question: {question}

Model answer: {answer}

`## Output Format`

`True / False`

Prompt Template for P_{True}-Cand

System Prompt:

You are a rigorous fact-checking assistant. Given a question, a previously generated answer, and a list of candidate answers, you must judge whether the model's answer is factually correct. Respond strictly with either `True` or `False`—no explanations, no additional text.

User Prompt:

`## Instruction`

You will be given a question, the model's answer, and several candidate answers. Please respond with `True` if the model's answer is factually correct, or `False` if it is incorrect. Answer strictly with either "True" or "False" only.

`## Input`

Question: {question}

Model answer: {answer}

Candidate answers: {candidate_1}, {candidate_2}, {candidate_3}, . . .

`## Output Format`

`True / False`

Prompt Template for Self_{Ask}

System Prompt:

You are a calibrated evaluator. After seeing a question and the model's own answer, you must assess how confident the model should be that its answer is correct. Output only a single numeric value between 0 and 1—no explanations, no additional text.

User Prompt:

`## Input`

Question: {question}

Answer: {answer}

`## Instruction`

How confident are you that the above answer is correct? Please output only a number between 0 and 1 (e.g., 0.85 means 85% confident). Use two-decimal precision.

`## Output Format`

`{confidence between 0 and 1 two-decimal precision}`

Prompt Template for Self$_{\text{Ask}}$-Cand

System Prompt:

You are a calibrated evaluator. After seeing a question, the model's own answer, and a set of candidate answers, you must assess how confident the model should be that its original answer is correct. Output only a single numeric value between 0 and 1—no explanations, no additional text.

User Prompt:

```
## Input
```

Question: {question}
Model answer: {answer}
Candidate answers: {candidate_1}, {candidate_2}, {candidate_3}, . . .

```
## Instruction
```

Considering the candidate answers above, how confident are you that the model's original answer is correct? Please output only a number between 0 and 1 (e.g., 0.85 means 85% confident). Use two-decimal precision.

```
## Output Format
{confidence between 0 and 1
two-decimal precision}
```

Table 13: **AUROC score on Qwen-2.5-14B-Instruct** of different calibration methods under increasing answer mixture. $\tau = 0$ denotes no-filtering, while $\tau \neq 0$ uses the threshold value tuned on the development set. **Bold** marks the best, underlined the second-best in each column. † indicates the difference from our SCA method is not statistically significant ($p > 0.05$).

Method	[1]	[1,2]	[1,2,4]	[1,2,4,6]
Single-turn				
Question-level				
Prob Entropy	81.0	78.6	76.5	75.3
N-Prob Entropy	81.3	79.0	77.0	75.3
Sem Entropy	77.8	77.6	75.6	74.2
Answer-level				
Verb	69.1	65.6	62.3	60.1
Verb-Topk	60.6	57.0	53.5	51.5
Consis	78.6	78.0	76.4	75.2
Consis-Verb	74.1	72.6	71.4	69.9
Consis-Verb-Topk	78.1	76.5	74.6	73.5
Perplexity	81.6	79.8	77.6	76.0
Double-turn				
P_{True}-Consis	74.3	72.8	72.1	71.1
P_{True}-Prob	78.2	76.6	76.2	75.1
P_{True}-Consis-Cand	69.6	67.8	67.3	66.8
P_{True}-Prob-Cand	79.4	77.0	75.3	74.2
Self-Ask	67.0	65.7	65.4	64.8
Self-Ask-Cand	74.9	72.9	70.2	69.1
Confidence Aggregation Baselines				
SNCA ($\tau = 0.4$)	76.7	76.6	74.7	73.3
SNCA ($\tau = 0$)	50.0	50.0	50.0	50.0
SFCA ($\tau = 0.3$)	78.5	77.7	75.6	74.1
SFCA ($\tau = 0$)	50.0	50.0	50.0	50.0
Ours				
SCA ($\tau = 0.1$)	**82.7**	**81.0**	**78.7**	**77.1**
SCA ($\tau = 0$)	82.6	80.9	**78.7**	**77.1**

Table 14: **AUROC score on Qwen-2.5-32B-Instruct** of different calibration methods under increasing answer mixture. $\tau = 0$ denotes no-filtering, while $\tau \neq 0$ uses the threshold value tuned on the development set. **Bold** marks the best, underlined the second-best in each column. † indicates the difference from our SCA method is not statistically significant ($p > 0.05$).

Method	[1]	[1,2]	[1,2,4]	[1,2,4,6]
Single-turn				
Question-level				
Prob Entropy	77.9	74.2	73.5	72.0
N-Prob Entropy	78.3	74.6	73.5	72.0
Sem Entropy	73.9	70.5	70.5	69.5
Answer-level				
Verb	62.8	60.3	59.4	58.7
Verb-Topk	58.9	55.2	51.6	49.3
Consis	75.3	72.3	72.4	71.3
Consis-Verb	73.7	70.7	69.8	68.7
Consis-Verb-Topk	75.5	73.7	72.8	71.4
Perplexity	**79.2**	75.8	74.7	73.1
Double-turn				
P_{True}-Consis	69.5	70.5	69.8	69.7
P_{True}-Prob	79.0	**78.2**	**77.3**	**77.0**
P_{True}-Consis-Cand	66.8	67.0	66.4	66.4
P_{True}-Prob-Cand	78.3	76.4	75.0	74.8
Self-Ask	72.4	70.5	68.3	67.1
Self-Ask-Cand	72.6	69.8	67.5	66.4
Confidence Aggregation Baselines				
SNCA ($\tau = 0.65$)	72.9	69.1	69.3	68.3
SNCA ($\tau = 0$)	50.0	50.0	50.0	50.0
SFCA ($\tau = 0.35$)	73.8	71.9	71.5	70.3
SFCA ($\tau = 0$)	50.0	50.0	50.0	50.0
Ours				
SCA ($\tau = 0.05$)	**79.2**	76.1	75.3	73.8
SCA ($\tau = 0$)	79.1	76.1	75.3	73.8

Table 15: **AUROC score on Qwen-2.5-72B-Instruct** of different calibration methods under increasing answer mixture. $\tau = 0$ denotes no-filtering, while $\tau \neq 0$ uses the threshold value tuned on the development set. **Bold** marks the best, underlined the second-best in each column. † indicates the difference from our SCA method is not statistically significant ($p > 0.05$).

Method	[1]	[1,2]	[1,2,4]	[1,2,4,6]
Single-turn				
Question-level				
Prob Entropy	**82.4**†	79.1	78.4	77.5
N-Prob Entropy	81.8†	78.4	77.0	76.2
Sem Entropy	77.3	75.6	75.1	73.9
Answer-level				
Verb	64.2	60.6	58.5	57.4
Verb-Topk	67.7	64.1	59.9	57.7
Consis	78.2	77.1	76.7	75.8
Consis-Verb	77.5	75.8	75.4	74.1
Consis-Verb-Topk	75.1	73.2	72.2	71.0
Perplexity	82.0†	79.9†	78.5	77.4
Double-turn				
P_{True}-Consis	73.7	70.0	69.4	69.6
P_{True}-Prob	81.3†	77.2	76.2	76.3
P_{True}-Consis-Cand	71.1	69.2	68.3	68.2
P_{True}-Prob-Cand	81.5†	77.5	75.8	75.6
Self-Ask	67.1	64.1	62.1	61.6
Self-Ask-Cand	74.4	70.5	67.2	66.0
Confidence Aggregation Baselines				
SNCA ($\tau = 0.2$)	76.5	74.9	73.5	72.0
SNCA ($\tau = 0$)	50.0	50.0	50.0	50.0
SFCA ($\tau = 0.15$)	76.4	75.2	74.2	73.4
SFCA ($\tau = 0$)	50.0	50.0	50.0	50.0
Ours				
SCA ($\tau = 0.3$)	82.0	80.3	**79.5**	**78.4**
SCA ($\tau = 0$)	82.0	**80.4**	**79.5**	**78.4**

Table 16: **AUROC score on LLaMA-3.1-8B-Instruct** of different calibration methods under increasing answer mixture. $\tau = 0$ denotes no-filtering, while $\tau \neq 0$ uses the threshold value tuned on the development set. **Bold** marks the best, underlined the second-best in each column. † indicates the difference from our SCA method is not statistically significant ($p > 0.05$).

Method	[1]	[1,2]	[1,2,4]	[1,2,4,6]
Single-turn				
Question-level				
Prob Entropy	70.3	68.2	63.8	61.5
N-Prob Entropy	65.6	63.2	60.1	58.1
Sem Entropy	78.5	77.9	74.8	73.1
Answer-level				
Verb	56.6	56.9	55.1	54.4
Verb-Topk	56.6	55.8	54.3	53.3
Consis	81.5†	80.7†	77.4	77.0
Consis-Verb	77.4	76.6	74.7	74.2
Consis-Verb-Topk	77.8	76.6	74.8	74.1
Perplexity	80.8†	79.3	76.2	75.4
Double-turn				
P_{True}-Consis	73.5	71.1	68.1	67.6
P_{True}-Prob	75.9	72.5	69.0	68.5
P_{True}-Consis-Cand	73.8	71.2	68.6	67.8
P_{True}-Prob-Cand	74.3	71.7	69.3	68.5
Self-Ask	67.6	64.5	63.0	63.2
Self-Ask-Cand	77.7	75.6	73.6	72.9
Confidence Aggregation Baselines				
SNCA ($\tau = 0.2$)	75.4	74.9	72.3	70.5
SNCA ($\tau = 0$)	50.0	50.0	50.0	50.0
SFCA ($\tau = 0.1$)	77.8	76.9	74.7	74.9
SFCA ($\tau = 0$)	50.0	50.0	50.0	50.0
Ours				
SCA ($\tau = 0.05$)	**81.6**	81.3	**78.3**	77.7
SCA ($\tau = 0$)	**81.6**	**81.4**	**78.3**	**77.8**

Table 17: **Confidence scores of Qwen2.5-7B-Instruct** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Prob Entropy	67.8	68.2	66.9	66.6
N-Prob Entropy	92.1	92.3	92.0	91.9
Sem Entropy	64.7	65.5	63.6	62.9
Answer-level				
Verb	94.7	95.4	94.8	94.7
$Verb_{Topk}$	82.5	81.3	73.7	71.8
Consistency	53.1	54.7	52.6	51.9
Consis-Verb	44.9	44.3	41.2	40.5
Consis-Verb-Topk	35.1	35.3	31.1	30.7
Perplexity	79.2	79.6	79.4	79.1
Double-turn				
P_{True}-Consis	46.9	47.0	45.0	46.9
P_{True}-Prob	46.8	46.9	44.9	46.8
P_{True}-Consis-Cand	59.3	61.0	62.9	65.0
P_{True}-Prob-Cand	59.4	61.1	63.0	65.2
$Self_{Ask}$	73.7	73.8	73.0	73.7
$Self_{Ask}$-Cand	70.5	71.5	71.1	72.2

Table 18: **Confidence scores of Qwen2.5-14B-Instruct** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Prob Entropy	73.4	73.3	72.4	71.7
N-Prob Entropy	93.7	93.6	93.5	93.2
Sem Entropy	70.6	71.3	69.9	68.4
Answer-level				
Verb	87.4	87.1	85.6	85.4
$Verb_{Topk}$	87.6	84.0	74.1	70.5
Consistency	62.7	63.3	62.5	61.0
Consis-Verb	42.2	45.6	43.4	41.7
Consis-Verb-Topk	45.5	44.8	42.7	40.7
Perplexity	85.6	85.8	85.8	85.4
Double-turn				
P_{True}-Consis	50.1	52.2	49.4	49.5
P_{True}-Prob	50.1	52.2	49.4	49.5
P_{True}-Consis-Cand	67.1	69.8	71.8	71.9
P_{True}-Prob-Cand	67.1	69.8	71.8	71.9
$Self_{Ask}$	77.5	78.7	76.0	75.2
$Self_{Ask}$-Cand	76.2	78.2	78.0	77.4

Table 19: **Confidence scores of Qwen2.5-32B-Instruct** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Prob Entropy	74.9	74.2	73.4	72.7
N-Prob Entropy	94.1	94.1	93.8	93.6
Sem Entropy	72.8	72.4	71.2	69.1
Answer-level				
Verb	88.6	88.1	87.3	87.7
$Verb_{Topk}$	87.6	82.4	75.1	72.4
Consistency	64.9	63.9	62.9	61.2
Consis-Verb	54.1	52.7	51.8	50.5
Consis-Verb-Topk	48.1	45.9	44.7	42.2
Perplexity	85.9	85.7	85.5	85.0
Double-turn				
P_{True}-Consis	69.4	76.2	78.2	78.4
P_{True}-Prob	69.5	76.2	78.3	78.5
P_{True}-Consis-Cand	74.9	78.7	81.0	81.9
P_{True}-Prob-Cand	74.9	78.7	81.1	81.9
$Self_{Ask}$	72.8	72.5	70.7	72.4
$Self_{Ask}$-Cand	73.2	75.2	75.6	76.1

Table 20: **Confidence scores of Qwen2.5-72B-Instruct** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Prob Entropy	76.4	75.4	74.3	73.4
N-Prob Entropy	94.1	93.9	93.5	93.3
Sem Entropy	71.8	70.3	68.3	66.6
Answer-level				
Verb	87.9	87.3	86.2	86.8
$\text{Verb}_{\text{Topk}}$	85.9	79.5	70.8	69.9
Consistency	65.4	64.2	61.5	59.8
Consis-Verb	59.0	58.2	55.5	53.7
Consis-Verb-Topk	53.2	52.8	48.1	45.0
Perplexity	87.7	87.2	86.2	85.5
Double-turn				
P_{True}-Consis	75.1	73.3	68.6	73.7
P_{True}-Prob	75.2	73.3	68.8	73.9
P_{True}-Consis-Cand	75.8	76.0	73.4	77.6
P_{True}-Prob-Cand	75.8	76.1	73.5	77.7
Self_{Ask}	84.9	79.7	72.9	75.3
Self_{Ask}-Cand	83.5	81.8	76.6	79.5

Table 21: **Confidence scores of LLaMA-3.1-8B-Instruct** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Prob Entropy	63.4	62.5	60.6	59.5
N-Prob Entropy	89.7	89.3	88.8	88.6
Sem Entropy	49.9	51.4	47.5	45.1
Answer-level				
Verb	96.2	96.1	96.2	96.2
$\text{Verb}_{\text{Topk}}$	95.8	96.3	96.2	96.3
Consistency	36.0	36.8	32.7	30.4
Consis-Verb	37.1	37.0	33.5	30.7
Consis-Verb-Topk	30.4	29.8	26.9	24.6
Perplexity	67.0	67.6	65.9	64.4
Double-turn				
P_{True}-Consis	53.3	50.8	46.2	45.9
P_{True}-Prob	53.1	50.9	46.4	45.7
P_{True}-Consis-Cand	64.8	65.7	66.1	65.7
P_{True}-Prob-Cand	64.7	65.7	65.9	65.9
Self_{Ask}	37.7	31.7	26.8	27.2
Self_{Ask}-Cand	45.4	47.8	46.3	46.1

Table 22: **Confidence scores of GPT-4o-mini** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Sem Entropy	49.6	50.1	46.2	44.9
Answer-level				
Verb	93.2	92.1	91.1	90.8
$\text{Verb}_{\text{Topk}}$	83.7	73.0	61.5	56.5
Consistency	60.1	60.3	57.6	56.5
Consis-Verb	54.1	54.3	51.2	50.2
Consis-Verb-Topk	48.7	46.3	42.4	41.1
Double-turn				
P_{True}-Consis	62.5	67.7	68.3	70.5
P_{True}-Consis-Cand	72.3	77.0	77.6	79.1
Self_{Ask}	76.3	76.1	73.7	74.9
Self_{Ask}-Cand	73.7	76.4	76.9	76.6

Table 23: **Confidence scores of GPT-4o** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Sem Entropy	63.0	54.1	45.4	40.4
Answer-level				
Verb	92.1	90.7	88.8	88.5
$\text{Verb}_{\text{Topk}}$	89.6	85.6	71.8	67.4
Consistency	70.1	64.3	56.5	52.6
Consis-Verb	68.8	62.8	54.8	50.6
Consis-Verb-Topk	65.1	58.5	48.8	45.8
Double-turn				
P_{True}-Consis	64.3	66.8	68.4	70.4
P_{True}-Consis-Cand	80.0	81.0	80.5	80.0
Self_{Ask}	77.8	75.8	74.9	75.3
Self_{Ask}-Cand	80.5	80.4	79.0	77.9

Table 24: **Confidence scores of DeepSeek-V3** under different ground truth(GT) set sizes.

Method	1a	2a	4a	6a
Single-turn				
Question-level				
Sem Entropy	80.9	78.6	76.2	72.9
Answer-level				
Verb	85.3	85.5	85.8	86.0
$\text{Verb}_{\text{Topk}}$	63.5	57.5	52.7	50.5
Consistency	81.1	79.1	76.7	73.2
Consis-Verb	69.0	65.9	60.6	55.6
Consis-Verb-Topk	63.4	58.5	53.6	49.0
Double-turn				
P_{True}-Consis	64.3	64.6	64.7	66.2
P_{True}-Consis-Cand	82.4	81.5	82.5	80.5
Self_{Ask}	76.5	72.8	69.8	71.2
Self_{Ask}-Cand	83.8	81.3	80.8	78.7