DISPERSION IS NOT DOUBT: RETHINKING LLM CONFIDENCE ESTIMATION WHEN MULTIPLE ANSWERS ARE VALID

Anonymous authors
Paper under double-blind review

ABSTRACT

Confidence estimators often treat concentrated support as evidence of correctness, relying—explicitly or implicitly—on the assumption that concentrated support implies a correct answer. When several answers are valid, however, this assumption fails: support can spread across correct alternatives without indicating error. We call this mismatch the *concentration–correctness gap*, and propose two layered conjectures. *(a) The inversion conjecture:* as the number of valid answers K grows, confidence falls even as the model answers better. *(b) The dispersion-is-not-doubt conjecture:* the decline stems from support being re-divided among valid answers, not from greater uncertainty about correctness. To test them, we introduce **MACE** (**M**ulti-**A**nswer **C**onfidence **E**stimation), $12{,}000$ open-ended questions with controlled answer cardinalities ($K = 1/2/4/6$) and exhaustive, expert-audited reference sets. Across 9 models and 15 training-free estimators, both conjectures are confirmed: a multi-answer confidence inversion is widespread; all 15 estimators lose AUROC under pooled cardinalities; and valid mass stays stable on correctly answered questions while support is unevenly re-divided—entropy decomposition, elicited confidence, and cross-model analysis ($r = 0.93$) jointly attribute the decline to valid-answer diversity. These findings motivate **CARE** (**CA**rdinality-**R**escaled **E**vidence), which rescales sampled answer frequency by question-level cardinality to approximate the total support on valid answers. Using verified counts, CARE raises AUROC from 74.6 to 78.9 on the broadest mixture, recovering approximately 77% of the loss and demonstrating the value of estimating the number of valid answers for more reliable confidence.

1 INTRODUCTION

Large language models are increasingly expected to indicate when their answers can be trusted (Yuchi et al., 2026; Hochman et al., 2026). Reliable confidence scores help determine when to retrieve additional evidence, verify an answer, or abstain (Maynez et al., 2020; Min et al., 2023; Ni et al., 2024). Training-free estimators are particularly attractive because they require neither fine-tuning nor labeled confidence annotations. They draw on four main sources: *(i)* uncertainty over the whole predictive distribution, including token or semantic entropy (Malinin & Gales, 2020; Kuhn et al., 2023); *(ii)* support for the returned answer, measured through sequence likelihood or agreement across samples (Jiang et al., 2021; Si et al., 2022; Manakul et al., 2023; Zhang et al., 2023); *(iii)* self-reported confidence (Lin et al., 2022; Xiong et al., 2023); and *(iv)* separate correctness verification (Kadavath et al., 2022; Tian et al., 2023).

Despite their differences, many of these signals rely—explicitly (*i* and *ii*) or implicitly (*iii* and *iv*)—on a shared implicit assumption: the more concentrated the support, the more likely the returned answer is correct; conversely, dispersed support is read as doubt. When a question admits only one valid answer, this assumption has a natural basis: support assigned to that answer is support for correctness, while mass on semantically distinct alternatives necessarily supports errors.

The Concentration–correctness Gap. When several answers are valid, however, this assumption breaks down. A question asking for a country a river flows through, or a past winner of an award, admits multiple correct responses, so probability mass outside the returned answer may support

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

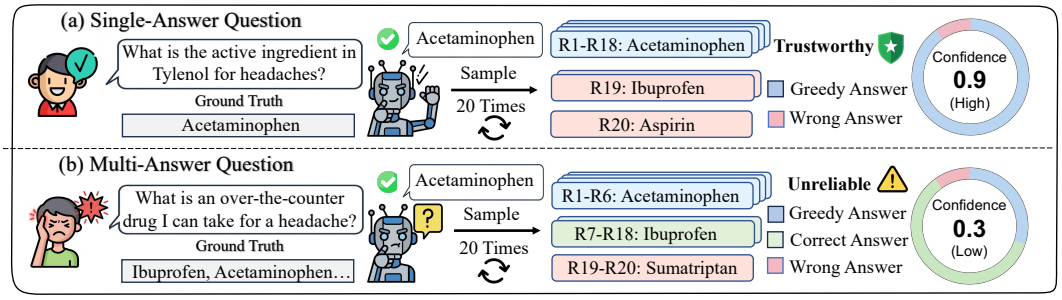

Figure 1: **The concentration–correctness gap between predictive concentration and answer correctness.** For a single-answer question, being correct and concentrating on one answer coincide, so a concentration-based score doubles as a correctness signal. For a multi-answer question the two come apart: equally correct responses spread across several valid answers, so concentration falls while correctness holds. The score then stops tracking whether the generated answer is correct.

another correct response rather than an error. Concentration and correctness then come apart: confidence tracks concentration, which no longer equals correctness. We call this mismatch the **concentration–correctness gap** (Figure 1), and propose two layered conjectures. **(a) The inversion conjecture:** as the number of valid answers K grows, confidence will fall even as the model answers better—confidence and correctness move in opposite directions. **(b) The dispersion-is-not-doubt conjecture:** the decline reflects support being re-divided among multiple valid answers, not greater uncertainty about correctness—dispersion is not doubt. Going further, the gap forces a rethink of confidence estimation: *a good estimator should reflect answer correctness faithfully, no matter how many answers are valid.*

Testing these conjectures requires a more complete benchmark. We generalize single-answer evaluation to the general case: a question may admit K valid answers ($K \geq 1$). Existing benchmarks cannot serve this need—single-answer benchmarks are merely the special case $K = 1$, multiple-choice ones constrain the response space, and open-ended ones neither control K nor provide complete reference sets. We therefore introduce **MACE** (**M**ulti-**A**nswer **C**onfidence **E**stimation), comprising 12,000 factual questions across six domains and four cardinality levels K. Within each question group, the knowledge type and sentence frame remain fixed while a constraint is progressively widened to admit more valid answers. Each question requests one real-world entity without supplying candidate options and has a finite, fully enumerated reference set. Domain experts audit both the questions and their answers for validity and completeness.

Evaluating 9 models and 15 training-free estimators on MACE, we first confirm conjecture (a): as K grows, confidence decreases even as accuracy improves—a **multi-answer confidence inversion**. A corollary of (a) follows directly: once questions of different cardinalities share one confidence scale—the norm in real deployment—confidence should become less able to distinguish correct from incorrect answers as the pool widens. Pooling questions exactly this way, we find that on LLaMA-3.1-70B-Instruct, all 15 estimators lose AUROC from the single-answer pool to the broadest mixture, with 13 declining by more than one point. The degradation also affects selective prediction: when Consis retains only the highest-scoring half of each pool, its accuracy gain over answering all questions falls from 20.1 to 13.8 points. The effect thus goes beyond lower confidence to weaker identification of trustworthy answers.

To test conjecture (b), we examine how valid probability mass is distributed. Among correctly answered questions, total mass on valid answers (the mass summed over the K valid answers) stays approximately stable as K increases, but the returned answer's share falls; the redistribution is uneven, with the returned answer retaining most of the valid mass while the other $K-1$ alternatives divide a growing remainder. Estimators are affected through explicit and implicit pathways. **Explicitly:** whole-distribution entropy rises with K—a decomposition attributes most of the rise to valid-answer diversity—while returned-answer likelihood and sample agreement fall as support shifts to other correct alternatives. **Implicitly:** Verb-Topk elicits confidence across competing candidates: total confidence on valid answers rises but the returned answer's share falls, with the largest single AUROC decline. Even separate verification, which assesses validity rather than dominance, mostly loses

discrimination once cardinalities are mixed. The effect also grows with model knowledge: across six models, larger growth in valid-answer entropy tracks larger confidence declines ($r = 0.93$), and larger models within a family fall more despite answering more accurately—broader coverage gives support more valid answers to spread across. Conjecture (b) is thus strongly supported: the decline tracks diversity among valid answers, not uncertainty about correctness.

This diagnosis suggests that, if the number of valid answers K were known in advance, one could use the returned answer's frequency to approximately estimate the total support on valid answers, escaping the reliance on concentration and estimating confidence more faithfully. We therefore propose **CARE** (**CA**rdinality-**R**escaled **E**vidence), which rescales sampled answer frequency by K. CARE leaves single-answer scores unchanged and preserves within-cardinality rankings, adjusting only comparisons across cardinalities. Using verified counts, it improves AUROC from 74.6 for Consis to 78.9 on the broadest mixture, reducing the decline from 5.6 to 1.3 points and recovering approximately 77% of the loss. These results take a first step toward the standard stated above and establish the value of cardinality information; obtaining the count itself—predicting the number of valid answers from the question and its validity constraints—is left to future work.

Our contributions are as follows:

- **The concentration–correctness gap.** We show that the "concentrated support implies correctness" assumption behind training-free estimators fails when multiple answers are valid, and we formulate and verify two conjectures on 9 models and 15 estimators: confidence falls as K grows while accuracy rises, and it stems from support re-divided among valid answers, not from doubt.
- **MACE.** $12{,}000$ open-ended factual questions across six domains and four cardinality levels ($K = 1/2/4/6$), with matched question designs within each group and fully enumerated, expert-audited reference sets.
- **CARE.** Rescaling sampled answer frequency by K recovers approximately 77% of Consis's AUROC loss on the broadest mixture ($74.6 \rightarrow 78.9$), demonstrating the value of cardinality information and motivating question-based cardinality estimation.

2 Related Work

Confidence and uncertainty estimation for LLMs. Training-based methods use supervision or fine-tuning to improve how a model expresses uncertainty (Lin et al., 2022; Zhang et al., 2024; Yang et al., 2024b; Ni et al., 2025b), at the cost of labels and with uncertain transfer (Ni et al., 2025a). Training-free methods instead read a score off the model at inference time, and they differ in which quantity they read. **Question-level** scores are functionals of the model's whole output distribution for the question: predictive entropy over sampled sequences (Malinin & Gales, 2020), its length-normalized form (Kadavath et al., 2022), and semantic entropy, which first merges outputs that mean the same thing (Kuhn et al., 2023). **Answer-level** scores attach to the response that was returned. Some are coupled to the generation, reading either the probability the model assigned to the emitted sequence (Guo et al., 2017; Jiang et al., 2021; Si et al., 2022) or the share of independently sampled responses that agree with it (Manakul et al., 2023; Cole et al., 2023; Lyu et al., 2025), with hybrids weighting that agreement by a self-reported number (Xiong et al., 2023). Others ask the model rather than its probabilities, either for a verbalized confidence stated beside the answer (Lin et al., 2022; Tian et al., 2023; Tanneru et al., 2024) or through a follow-up query that has it judge the answer already produced (Kadavath et al., 2022; Tian et al., 2023); hidden states and logits have also been used as features (Ren et al., 2022; Ling et al., 2024). On benchmarks with exactly one valid answer, dispersion and error coincide; open-ended multi-answer benchmarks are needed to expose the gap between them.

Factual QA benchmarks. Single-answer benchmarks such as TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019) and Mintaka (Sen et al., 2022) use open-ended responses and therefore satisfy the generation-format requirement, but hold cardinality at one and exclude the multi-answer phenomenon by construction. Multiple-choice benchmarks (Wang et al., 2024; Clark et al., 2018; Xu et al., 2025) confine outputs to a given option set, so they do not satisfy open-ended generation and suppress dispersion over the free-form response space. Open-ended multi-answer benchmarks (Berant et al., 2013; Talmor & Berant, 2018) allow the phenomenon but draw answers from incomplete knowledge bases, with cardinality varying by collection rather than by design. No

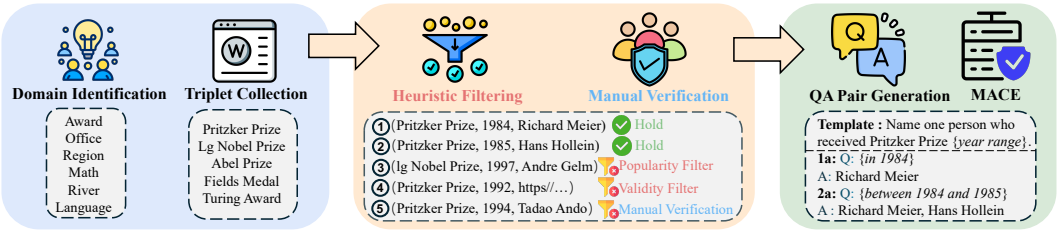

Figure 2: **The MACE benchmark construction pipeline**, illustrated using the *Award* domain. ①② **Valid triplets** (e.g., Pritzker Prize winners) are retained. ③ **Low-popularity triplets** (e.g., obscure awards like the Ig Nobel Prize) are removed via *Popularity Filter*. ④ **Noisy triplets** containing 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).

benchmark jointly provides *controlled answer cardinality*, *open-ended generation* and *finite and complete answer sets* (Appendix A).

3 The MACE Benchmark

In this section, we introduce MACE (**M**ulti-**A**nswer **C**onfidence **E**stimation), a diagnostic benchmark built to measure whether answer cardinality changes the reliability of confidence as a signal of correctness. We first state the requirements this purpose imposes (§3.1), then describe how the benchmark is concretely built (§3.2), and finally summarize the characteristics of MACE (§3.3).

3.1 Design Requirements

Attributing a change in confidence reliability to the number of valid answers requires three properties to hold jointly. **(P1) Controlled answer cardinality.** The number of valid answers must be the only variable across cardinality levels, with other factors, including knowledge type, question phrasing, and difficulty, controlled so that any observed change can be attributed to cardinality. **(P2) Open-ended generation.** Answers must be generated freely rather than selected from a fixed option set, allowing probability mass to spread over valid alternatives as they do in deployment. This requirement concerns the response format and does not require every question to have multiple valid answers. **(P3) Finite and complete answer sets.** The ground-truth set must contain all and only valid answers, so that no correct answer is omitted and no incorrect answer is included. No existing benchmark satisfies P1–P3 jointly (§2), so we take the three as the guiding principles for construction.

3.2 Benchmark Construction

Overview. MACE spans six factual domains: Honorable Award (Award), Political Office (Office), Regional Affiliation (Region), Mathematical Concept (Math), Natural River (River), and Linguistic Culture (Language). Each domain contains four question types with exactly 1, 2, 4, or 6 valid answers (1a, 2a, 4a, 6a) and 500 questions each, for 2,000 per domain and 12,000 in total (Table 5). Construction proceeds in three phases (Figure 2); full details are in Appendix §B.

Knowledge Collection. The choice of domain decides whether complete answer sets are attainable at all, since a question such as "name a great scientist" has an answer set that is subjective and impossible to close. *Domain Identification* therefore targets domains whose complete answer set can be enumerated objectively, using LLM-assisted brainstorming followed by manual screening to balance coverage against annotation cost (Appendix §B.1). *Triplet Collection* then gathers knowledge as (subject, relation, object) triplets: the Math domain generates them from arithmetic rules, and the other five retrieve them from Wikidata[1] under predefined relations, first fetching subjects and then their associated objects, following prior practice (Sen et al., 2022; Yuksekgonul et al., 2023; Mallen et al., 2023). Domain schemas are listed in Table 7.

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

Knowledge Filtering. Retrieved triplets are incomplete and noisy, so this phase trades scale for reliability in two stages. *Heuristic Filtering* discards obscure entities that no model could reasonably be expected to know, through a *Popularity Filter* over 12-month Wikipedia pageviews, and malformed or unresolvable entries, through a *Validity Filter* over predefined format rules; together these remove 92.4% of the collected triplets and bring the remainder into a range that can be checked by hand (Appendix §B.3). *Manual Verification* then has domain experts audit every remaining triplet for correctness and completeness of the answer set, correcting entries where possible and removing a further 19.05% of subjects. Table 6 reports the counts retained at each step.

QA Pair Generation. Following prior work (Mallen et al., 2023; Yuksekgonul et al., 2023), verified triplets become questions built from a subject and its relation, with the associated objects as the answer set. All domains share the sentence frame "Name one X that satisfies Y", which holds phrasing constant across cardinalities and asks for a single answer in the form a deployed system would return. Cardinality is controlled through the parameterized constraint Y: widening a year range or a numeric interval admits more valid answers while the frame, the relation, and the knowledge type stay fixed, as in "Name one person who received {award} between {lower_year} and {upper_year}", so questions at 1a and 6a differ in how many answers satisfy them and not in what they ask about. The `Const-Q` column of Table 5 reports how many questions this yields; we retain 500 per cardinality per domain by clustering-based stratified sampling, so that no domain or individual subject dominates the evaluation. Templates and the constraint-relaxation procedure are given in Appendix §B.5.

3.3 Data Quality

MACE combines four properties that existing benchmarks do not offer together.

- **Controlled answer cardinality (P1).** Within a domain, questions of every cardinality share one knowledge type, sentence frame, and answer type, differing only in the parameterized constraint, so cardinality varies while other factors stay comparable.
- **Open-ended rather than closed-form questions (P2).** Answers are real-world entities elicited without candidate options, so the model must recall them rather than select from a supplied set.
- **Finite and complete answer sets (P3).** Every triplet is verified by two independent annotators with a third adjudicating disagreements (Cohen's $\kappa = 0.94$), and stratified spot-checks are repeated until a full round surfaces no errors (Appendix §B.4).
- **Diverse domains and answer types.** MACE spans six domains across social, geographic, linguistic, and mathematical knowledge, covering heterogeneous factual relations, answer types, and difficulty levels. This breadth enables systematic evaluation of cardinality effects.

4 Experimental Setup

In this section, we state what a confidence score is evaluated against, which estimators we evaluate and how we group them, which models we run, and the notation used for the distributional quantities our analysis relies on. Full formulations, prompts, and inference settings are given in Appendix §C.

Evaluation target. The model answers each question by greedy decoding, and an answer is correct if it appears in the verified answer set. Every estimator returns one score expressing how confident the model is in that answer, and we report three quantities. **Accuracy** is how often the model is right, and **Confidence** is the mean score the estimator assigns to the returned answer. A deployed system must also separate correct answers from incorrect ones on a single scale, since a query does not announce how many answers are valid. To simulate increasingly complex real-world settings where questions with different answer cardinalities coexist, we report **AUROC** (Hanley & McNeil, 1982), the probability that a correct answer is ranked above an incorrect one, under the cumulative mixtures $[1]$, $[1, 2]$, $[1, 2, 4]$ and $[1, 2, 4, 6]$. AUROC is our primary metric, as it matches how confidence is consumed downstream, where the ordering among queries decides which to retrieve for, verify, or refuse. Calibration error does not apply to most of these estimators and is misleading here, moving opposite to AUROC as cardinality grows (Appendix §D.7). All metrics are macro-averaged over the six domains with equal weight (Appendix §C.2).

Estimators. Following the grouping of §2, we evaluate 15 estimators from prior work, organized by the quantity each score reads. **(i) The whole predictive distribution:** *Prob Entropy*, its length-normalized variant *N-Prob Entropy*, and *Sem Entropy* over semantic clusters. **(ii) The support of the returned answer:** *Perplexity*, *Consis*, and the hybrids *Consis-Verb* and *Consis-Verb-Topk*, which weight a consistency count by a verbalized number. **(iii) The model's self-report about the returned answer:** *Verb*, which states a confidence for the single answer it gave, and *Verb-Topk*, which lists its five candidate answers with a confidence for each and uses the highest-confidence candidate as the returned answer. **(iv) A separate verification query:** P_{True} and *Self-Ask*, which pose a follow-up question asking the model to judge the answer it has already produced. Three suffixes distinguish their variants: *-Prob* reads the probability the model assigns to the token "true" in its verdict, *-Consis* instead takes the fraction of sampled verdicts that say true, and *-Cand* appends the model's own alternative candidates to the verification prompt so the judgment is made against them rather than against the answer alone. The three entropies and Perplexity are uncertainties rather than confidences, and we map them by $\text{conf} = e^{-H}$ so that larger always means more confident; the map is monotone, so it leaves AUROC untouched. Formulations and prompts are in Appendix §C.1.

Models. We evaluate Qwen2.5-Instruct (7B, 14B, 32B, 72B) (Yang et al., 2024a), LLaMA-3.1-Instruct (8B, 70B) (Grattafiori et al., 2024), DeepSeek-V3 (Liu et al., 2024), GPT-4o-mini, and GPT-4o (Hurst et al., 2024), spanning open-weight and API-based models across families and scales. Due to space constraints, we present LLaMA-3.1-70B-Instruct as the representative model and trends are consistent across families, scales, and domains (Appendix § D.1, § C.3).

Notation. To study how support is distributed among valid answers and how this affects confidence, we distinguish total valid probability mass from the share assigned to each answer. Let $\pi_{\text{samp}}(\cdot \mid x)$ denote the empirical distribution of the $N = 20$ sampled responses after semantic grouping; each mode's probability is its sample frequency. For quantities based on sequence probabilities (notably Sem Entropy), we instead use the normalized cluster distribution π_{seq} defined in Appendix §C.1. Let $\mathcal{A}(x)$ be the verified answer set, with cardinality $k = |\mathcal{A}(x)|$. The **valid mass**, $q(x) = \sum_{a \in \mathcal{A}(x)} \pi_{\text{seq}}(a \mid x)$, measures sequence-probability support for correct answers, whereas sampled accuracy is the fraction of sampled responses that are valid. For $q(x) > 0$, each valid answer a_j receives sequence-probability mass $\pi_{\text{seq}}(a_j \mid x) = q(x) r_j(x)$, where its **share** $r_j(x) \in [0, 1]$ satisfies $\sum_{j=1}^{k} r_j(x) = 1$. When the returned answer is correct, we index it as a_1, so its support is qr_1. Finally, M^+ denotes the number of distinct valid answers observed among the N samples.

5 Results and Analysis

In this section, we examine how answer cardinality affects confidence and its usefulness for predicting correctness. We first test the **inversion conjecture** and examine the redistribution of valid-answer support (§5.1), then test its corollary under mixed cardinalities (§5.2), and finally analyze why sensitivity differs across estimators and models (§5.3)—which, together with the redistribution, supports the **dispersion-is-not-doubt conjecture**.

5.1 Correctness and Concentration Come Apart

Confidence falls as more answers become valid, while correctness rises. Table 1 reports each estimator's average score at cardinality 1, 2, 4 and 6, and all 15 are lower at 6 than at 1: Consis drops from 51.3 to 35.9, Sem Entropy from 60.8 to 45.2, Verb-Topk from 81.9 to 56.9, Perplexity from 77.6 to 71.0. The decline does not depend on which signal a score reads, and it recurs across domains, model families and parameter scales (Appendix §D.1, §C.3). Over the same range the model answers *better*: accuracy rises from 48.0 to 61.7, and not as an artifact of greedy decoding, since the share of sampled responses that are valid rises from 30.6 to 45.0 and the valid mass q from 47.5 to 62.0. It also surfaces more of the answer set as that set widens, covering $M^+ = 0.89$ distinct valid answers at cardinality 1 against 3.01 at cardinality 6 (Table 2, upper panel). Every quantity that tracks what the model knows therefore moves up while every score meant to report that knowledge moves down, and we call this opposite movement the **multi-answer confidence inversion**.

The valid mass is re-divided rather than reduced. To see what the scores respond to, we restrict attention to the questions the model answers correctly, where confidence ought to stay high, and

Table 1: **Average confidence score under each answer cardinality** for LLaMA-3.1-70B-Instruct. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups follow §4. The four scores marked e^{-H} are uncertainties mapped to confidence to display the trend.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Prob Ent.$^{e^{-H}}$	71.6	70.0	66.6	65.6	-6.0
N-Prob Ent.$^{e^{-H}}$	91.5	90.9	89.9	89.7	-1.8
Sem Ent.$^{e^{-H}}$	60.8	57.9	49.7	45.2	-15.6
(ii) Support of the returned answer					
Perplexity$^{e^{-H}}$	77.6	76.4	73.0	71.0	-6.6
Consis	51.3	48.9	40.2	35.9	-15.4
Consis-Verb	50.3	47.9	40.5	36.4	-13.9
Consis-Verb-Topk	47.6	45.9	38.4	34.2	-13.4
(iii) Self-report on the returned answer					
Verb	94.8	94.7	93.1	92.3	-2.5
Verb-Topk	81.9	73.9	62.1	56.9	-25.0
(iv) Separate verification query					
P_{True}-Consis	79.3	79.3	77.0	75.5	-3.8
P_{True}-Prob	79.2	79.0	76.8	75.4	-3.8
P_{True}-Consis-Cand	88.1	87.9	87.1	87.9	-0.2
P_{True}-Prob-Cand	88.1	87.8	87.1	87.8	-0.3
Self-Ask	63.6	60.9	56.1	54.4	-9.2
Self-Ask-Cand	64.5	63.5	59.6	56.8	-7.7

Table 2: **What the answer distribution does** as cardinality grows for LLaMA-3.1-70B-Instruct. We use $N = 20$ as the sampling setting. The upper panel covers all questions, the lower only those the model answers correctly. *greedy acc.* is the accuracy of the returned answer, *sampled acc.* the fraction of the 20 sampled responses that are valid, q the probability the model places on answers in the verified set, and M^+ the number of distinct valid answers among those samples. In the lower panel the share r_1 is a fraction of q rather than a probability, and q splits exactly into qr_1, the probability of the returned answer, and $q(1 - r_1)$, the probability on other valid answers.

	1a	2a	4a	6a	Δ
Whole answer distribution					
greedy acc.	0.480	0.554	0.599	0.617	$+0.137$
sampled acc.	0.306	0.362	0.418	0.450	$+0.144$
valid mass q	0.475	0.557	0.607	0.620	$+0.145$
M^+	0.89	1.46	2.42	3.01	$+2.12$
Correct answers only					
valid mass q	0.913	0.920	0.902	0.902	-0.011
share r_1	0.984	0.930	0.830	0.768	-0.216
qr_1	0.902	0.860	0.755	0.700	-0.202
$q(1 - r_1)$	0.011	0.060	0.147	0.202	$+0.191$

measure the distribution the estimators consume (Table 2, lower panel). The valid mass q is flat across cardinality (0.913, 0.920, 0.902, 0.902): on the questions it gets right, the model is exactly as right at cardinality 6 as at 1. What changes is how that same q is divided. The share held by the returned answer falls from $r_1 = 0.984$ to 0.768, moving mass from qr_1, which drops from 0.902 to 0.700, onto the model's other valid answers, which rise from 0.011 to 0.202. Nothing about the model's correctness moved; the evidence for it was spread over more answers.

The division is lopsided, which is why the decline is gradual. If k valid answers split the mass evenly, r_1 would fall as $1/k$: 0.50 at two valid answers and 0.17 at six. The table gives 0.930 and 0.768. The model keeps one strongly preferred answer and leaves the alternatives a remainder—even with six answers valid it still holds more than three quarters of the valid mass on the one it returns, the remaining 0.202 spread across all the others—so scores built on qr_1 declines gradually rather than halving. That mildness is why the effect is easy to overlook, and why the size of a confidence drop is a poor guide to the damage it does, which §5.2 measures directly.

5.2 The Consequence: Mixed Cardinalities Weaken Correctness Discrimination

Almost every estimator loses discrimination as the cardinality pool widens. Figure 3 evaluates AUROC as questions with different answer cardinalities are pooled. From $[1]$ to $[1, 2, 4, 6]$, all 15 estimators decline, 13 by more than one point; even the strongest score on MACE loses 5.6 points (Table 3). This pattern is consistent with the confidence inversion in §5.1: lower scores on more accurately answered multi-answer questions can impair correctness ranking across queries. The degradation also affects selective prediction. When answering only the highest-scoring half of each pool, Consis improves accuracy over answering all questions by 20.1 points at cardinality 1, but only 13.8 points in the broadest mixture—a reduction of nearly one third (Appendix §D.4).

The loss of discrimination varies substantially across estimator families. Whole-distribution estimators show the largest mean AUROC decline (-8.5 points), followed by self-report (-7.0), returned-answer support (-6.1), and verification queries (-3.2). These averages nevertheless mask substantial within-family variation. Verb-Topk exhibits the largest individual decline (-11.6 points, reaching 48.4 AUROC), whereas plain Verb changes modestly (-2.5) but starts from weak discrimination (58.0 at $[1]$). Verification queries also differ: P_{True}-Consis-Cand and Self-Ask-Cand remain

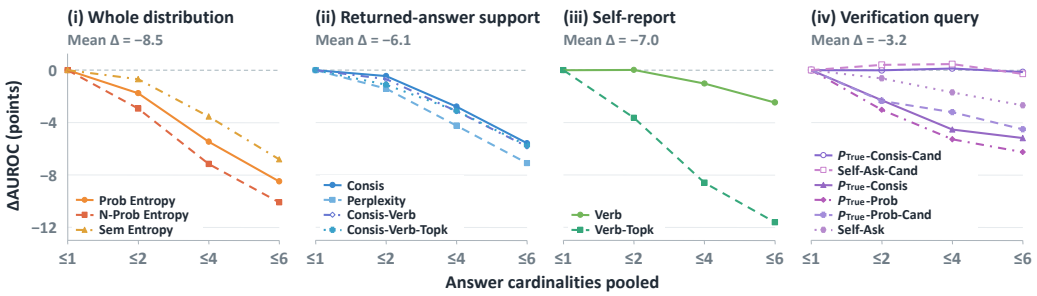

Figure 3: **Change in AUROC as answer cardinalities are pooled onto one confidence scale** (LLaMA-3.1-70B-Instruct). The horizontal axis gives the cumulative mixture: ≤ 4 pools questions with 1, 2 or 4 valid answers. The vertical axis is each estimator's AUROC relative to its own value on single-answer questions. Panels group the estimators by the quantity each reads (§4), annotated with the group's mean change from ≤ 1 to ≤ 6; the underlying absolute values are in Table 3.

nearly unchanged, while other variants, including P_{True}-Prob-Cand, decline more substantially. Neither self-report nor candidate-aware verification therefore provides uniform robustness to cardinality mixing. §5.3 examines the signals underlying these differences.

5.3 The Mechanism: Which Signals Fail, and Why

Explicit dependence: diversity and correctness enter the same score. Let π_V and π_I denote the conditional distributions over valid and invalid answer modes. Entropy decomposes as $H(\pi) = h(q) + qH(\pi_V) + (1-q)H(\pi_I)$, where the binary entropy $h(q)$ measures correctness uncertainty and the remaining terms capture diversity within each class (Appendix §D.9). Thus, entropy can rise even when total valid mass remains unchanged. Empirically, valid-answer diversity contributes $+0.313$ nats, exceeding the net entropy increase of $+0.284$ nats. Increased dispersion is also visible from the start of decoding: first-token entropy rises from 1.22 to 1.68 between cardinalities 1 and 6 (Table 30). Returned-answer scores are sensitive to the same redistribution: a correct answer's mass qr_1 decreases as its share r_1 falls, even at fixed q. Consistency measures support through sample agreement, Perplexity through sequence likelihood, and Consis-Verb hybrids through confidence-weighted agreement. These scores can therefore decline when support shifts toward other valid answers, without a corresponding reduction in total valid mass.

Implicit dependence: elicitation can preserve or reduce answer competition. Verb-Topk elicits alternatives and confidence for each. Its total reported confidence on valid answers rises ($0.45 \rightarrow 0.58$), but the returned answer's confidence falls ($0.81 \rightarrow 0.57$; Appendix §D.8), suggesting that verbalized confidence can inherit answer competition without reading generation probabilities. Plain Verb elicits no competing alternatives and changes less, although its discrimination remains weak. Separate verification instead assesses whether an answer is valid, potentially reducing dependence on its dominance. The stability of P_{True}-Consis-Cand and Self-Ask-Cand supports this interpretation, but the decline of P_{True}-Prob-Cand shows that candidate access alone does not ensure robustness.

Across models: broader knowledge coverage can amplify dilution. A model that knows more valid answers can distribute support across more correct alternatives. Its confidence in the returned answer may therefore decline more sharply, even when its overall accuracy is higher. Across six models, a larger increase in valid-answer entropy is associated with a greater confidence decline (Pearson $r = 0.93$; Appendix §D.10). Larger models within the same family also exhibit stronger declines despite answering more accurately (Figure 6). Together, these observations suggest that broader knowledge coverage can intensify dilution by expanding the set of valid answers.

6 CARE: Cardinality-Rescaled Evidence

In this section, we investigate whether question-level cardinality can help correct the distortion in answer-level confidence. We motivate **CARE** (**CA**rdinality-**R**escaled **E**vidence), introduce its rescaling rule and evaluate how much discrimination it recovers using verified answer counts.

Table 3: **AUROC under cumulative answer-cardinality mixtures** for LLaMA-3.1-70B-Instruct. Each column $\leq k$ pools questions with at most k valid answers; Δ denotes the change from $[1]$ to $[1, 2, 4, 6]$, computed before rounding. Estimator groups follow §4. **Bold** marks the best score and underlining the best existing estimator in each column. CARE additionally uses verified answer counts and equals Consis at ≤ 1 by construction. Results for other models appear in Appendix §D.6.

Method	≤ 1	≤ 2	≤ 4	≤ 6	Δ	Method	≤ 1	≤ 2	≤ 4	≤ 6	Δ
(i) Whole predictive distribution						*(iii) Self-report on the returned answer*					
Prob Entropy	73.8	72.0	68.3	65.3	−8.5	Verb	58.0	58.0	57.0	55.5	−2.5
N-Prob Entropy	72.6	69.7	65.5	62.5	−10.1	Verb-Topk	60.0	56.4	51.4	48.4	−11.6
Sem Entropy	78.9	78.2	75.3	72.1	−6.8	*(iv) Separate verification query*					
(ii) Support of the returned answer						P_{True}-Consis	68.4	66.1	63.9	63.3	−5.2
Consis	80.1	79.7	77.3	74.6	−5.6	P_{True}-Prob	72.5	69.4	67.2	66.2	−6.3
Perplexity	80.1	78.7	75.9	73.0	−7.1	P_{True}-Consis-Cand	62.5	62.5	62.6	62.4	−0.1
Consis-Verb	79.1	78.4	76.0	73.3	−5.8	P_{True}-Prob-Cand	75.1	72.8	71.9	70.6	−4.5
Consis-Verb-Topk	79.7	78.6	76.6	73.9	−5.8	Self-Ask	65.6	65.0	63.9	63.0	−2.7
						Self-Ask-Cand	65.9	66.3	66.4	65.6	−0.3
Answer-level evidence rescaled by verified cardinality (§6)											
CARE	**80.1**	**80.8**	**80.6**	**78.9**	**−1.3**						

Motivation. The intuition behind CARE is that the same answer frequency can represent different levels of support depending on how many answers are valid. When several alternatives are acceptable, a correct answer may receive less support simply because other valid answers also receive probability mass (§5.3). Its frequency alone can therefore understate the evidence for correctness relative to single-answer questions. Answer cardinality provides a reference for this potential dilution. Although the count cannot reveal how unevenly support is distributed, it can help make answer-level scores more comparable across questions.

Method. Let c be the fraction of $N = 20$ independently sampled responses that agree with the returned answer $\hat{y}$, and let $K = |\mathcal{A}(x)|$ be the number of valid answers. CARE assigns

$$S_{\mathrm{CARE}}(x, \hat{y}) = 1 - (1 - c)^K. \tag{1}$$

At the same frequency $0 < c < 1$, CARE assigns a higher score when more answers are valid, accounting for potential support dilution. It leaves single-answer scores unchanged when $K = 1$ and preserves rankings among questions with the same cardinality. For fixed K and small c, it approximates Kc, but saturates toward 1 as c grows. This bounded correction empirically outperforms direct multiplication (Appendix §D.12).

Results and implication. Using verified answer counts, CARE achieves the highest AUROC among the evaluated scores in every mixture containing multi-answer questions (Table 3). On the full mixture, it improves AUROC from 74.6 for Consis to 78.9, reducing the decline from 5.6 to 1.3 points and recovering approximately 77% of the loss. These results show that question-level cardinality provides useful information for interpreting answer-level evidence. While practical use requires estimating the count, the findings identify estimating answer cardinality from the question itself as a promising direction for mitigating the loss of correctness discrimination.

7 Conclusion

In this work, we identify the concentration–correctness gap: the assumption that concentrated support implies correctness fails when multiple answers are valid. On MACE, across nine models and 15 training-free estimators, we verify two conjectures: confidence falls despite higher accuracy as more answers become valid (the inversion conjecture), and the decline stems from support re-divided among valid answers, not correctness uncertainty (the dispersion-is-not-doubt conjecture); mixing cardinalities further weakens correctness discrimination. CARE then rescales sampled answer frequency by question-level cardinality, recovering approximately 77% of Consis's AUROC loss with verified counts—a first step toward confidence that reflects answer correctness no matter how many answers are valid and a motivation for question-based cardinality estimation.

AI USE STATEMENT

In this work, generative AI tools were used for language editing, manuscript organization, and literature-search assistance. GPT-4o was also used for domain brainstorming, answer evaluation, and semantic-equivalence judgments, as described in the appendices. All cited sources, technical claims, experimental procedures, code, analyses, and results were reviewed and verified by the authors. We take responsibility for the final content of this work.

ETHICS STATEMENT

MACE uses publicly available factual sources and rule-generated mathematical data, with human annotation for quality verification. The study does not collect private participant data or target sensitive personal attributes. Models and data sources are used for academic research under their applicable terms.

REPRODUCIBILITY STATEMENT

The appendices provide benchmark construction procedures, annotation protocols, question templates, estimator definitions, prompts, and inference settings. They also report detailed results across models, domains, and answer cardinalities, together with the derivations and implementation details supporting the main analysis.

REFERENCES

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

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

Jeremy Cole, Michael Zhang, Dan Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 530–543, 2023.

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

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

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

Hail Hochman, Natalie Shapira, and Yoav Goldberg. Factual retrieval in llms is a redundant, distributed and non-contiguous process. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 46747–46768, 2026.

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

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 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, 2021.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611, 2017.

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

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

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

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

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, et al. 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)*, pp. 3357–3370, 2024.

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

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 2511–2522, 2023.

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

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

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 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)*, pp. 9802–9822, 2023.

Potsawee Manakul, Adian Liusie, and Mark John Francis Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *ArXiv*, abs/2303.08896, 2023. URL `https://api.semanticscholar.org/CorpusID:257557820`.

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

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 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*, pp. 12076–12100, 2023.

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

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

Shiyu Ni, Keping Bi, Jiafeng Guo, Lulu Yu, Baolong Bi, and Xueqi Cheng. 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)*, pp. 24315–24329, 2025b.

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

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

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

Alon Talmor and Jonathan Berant. 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)*, pp. 641–651, 2018.

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

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 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*, pp. 5433–5442, 2023.

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

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

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

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, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yunyang Wan, Yuqi Liu, Zeyu Cui, Zhenru Zhang, Zihan Qiu, Shanghaoran Quan, and Zekun Wang. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024a. URL `https://api.semanticscholar.org/CorpusID:274859421`.

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

Henry Shaowu Yuchi, Michal Kucer, Benjamin H Sims, Selma Peterson, and Emily Taylor. Nuclearqav2: A structured benchmark for evaluating domain-science competence in large language models. *arXiv preprint arXiv:2606.27047*, 2026.

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

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 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)*, pp. 7113–7139, 2024.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 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*, pp. 15445–15458, 2023.

APPENDIX CONTENTS

A Why Existing Benchmarks Cannot Measure the Concentration–Correctness Gap **15**

B The MACE Benchmark **16**

B.1 Domain Identification 16

B.2 Triplet Collection 18

B.3 Heuristic-Based Filtering 18

B.4 Manual Verification 20

B.5 QA Pair Generation 20

C Experiment Setup **22**

C.1 Confidence Estimation Methods 22

C.2 Experiment Details 26

C.3 Domain Effects: Relation Between Question Type and Difficulty 29

D Results and Analysis **30**

D.1 Consistent Trends Across Models and Scales 30

D.2 Instruments Computed from the Verified Answer Sets 31

D.3 Discrimination Within a Single Cardinality 34

D.4 Selective Prediction 35

D.5 Pooled Degradation Is Not a Base-Rate Artifact 36

D.6 AUROC under Cardinality Mixtures Across Models 36

D.7 Expected Calibration Error 37

D.8 Why Does Confidence Decline? A Per-Method Analysis 40

D.9 Entropy Decomposition 43

D.10 Correlation Between Confidence and Knowledge Coverage 44

D.11 Analysis of Cluster Probability Distribution 45

D.12 The Answer Count: Correction Form, Required Accuracy, and Where It Could Come From 46

A WHY EXISTING BENCHMARKS CANNOT MEASURE THE CONCENTRATION–CORRECTNESS GAP

This appendix expands the benchmark comparison summarized in §2. Measuring the concentration–correctness gap in open-ended factual QA places stringent demands on the evaluation benchmark: the three properties of §3.1 must hold *jointly* (Table 4).

- **P1. Controlled Answer Cardinality.** Questions must be organized into systematic cardinality tiers (e.g. $|\mathcal{A}| \in \{1, 2, 4, 6\}$) built from comparable knowledge types and templates, so that observed changes in confidence reliability can be attributed to answer multiplicity rather than to shifting phrasing or knowledge type. Without this control, any effect is confounded.
- **P2. Open-Ended Generation.** The benchmark must contain free-form questions rather than a fixed candidate option set. This requirement concerns the response format, not the number of valid answers: it permits probability mass and sampled responses to spread over valid alternatives when they exist, while also covering the single-answer case as a necessary baseline.
- **P3. Finite and Complete Answer Sets.** For every question, the ground-truth answer set must enumerate *all and only* the correct answers. Without completeness, a model that generates a valid but unlisted answer is wrongly counted as an error, and since our entire analysis is expressed in terms of separating correct from incorrect predictions, such false negatives corrupt the measurement at its foundation.

We organize existing benchmarks by *answer-space structure*, the dimension most relevant to dispersion across valid answers, and show that none satisfies P1–P3 simultaneously.

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. This format permits clean correctness measurement, but the model's probability is confined to a predefined option set (e.g. A/B/C/D), bypassing the open-ended dispersion over the full vocabulary that characterizes free-form generation. They fail **P1** and **P2**.

Multi-correct multiple choice. SATA-Bench (Xu et al., 2025) adopts a "select all that apply" format, permitting multiple correct options per question (**P3** ✓). The model still operates over a closed, small option set rather than generating freely, which limits ecological validity for studying open-ended dispersion. More critically, the number of correct options varies arbitrarily rather than being controlled, so calibration or discrimination cannot be traced as a function of answer cardinality. This fails **P1**.

Open single-answer QA. 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 (**P3** ✓), and the free-form response format satisfies **P2**; however, dispersion across valid answers cannot occur, so these benchmarks fail **P1**. Such benchmarks are suitable for studying single-answer confidence but structurally incapable of diagnosing the multi-answer effect.

Open multi-answer QA (non-exhaustive). WebQuestions (Berant et al., 2013) and ComplexWebQuestions (Talmor & Berant, 2018) derive answers from structured knowledge bases and naturally yield some multi-entity answer sets (**P2** ✓). However, knowledge bases are systematically incomplete: entities and relations are frequently missing. The ground-truth answer sets are therefore *not* exhaustive—a model may produce a factually correct answer absent from the knowledge base—violating **P3**. This directly contaminates the evaluation, since correct predictions are counted as errors, and the resulting noise is itself correlated with answer multiplicity. Answer cardinality is also uncontrolled, failing **P1**.

In summary, closed-format benchmarks preclude open-ended dispersion, open single-answer benchmarks cannot exhibit the effect at all, and the only open multi-answer benchmarks draw answers from incomplete knowledge bases, so correct predictions are scored as errors at a rate that itself grows with answer multiplicity. Because no existing dataset provides controlled answer cardinality, open-ended generation, and finite and complete answer sets at the same time, none of them can separate the effect

Table 4: Comparison of existing benchmarks against the three properties of §3.1 required to measure how answer cardinality affects the reliability of confidence in open-ended factual QA. **P1**: controlled answer cardinality; **P2**: open-ended generation; **P3**: finite and complete answer sets. 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 dispersion across valid answers cannot occur.
Multi-answer (non-exhaustive)	WebQuestions, CWQ	✗	✓	✗	Incomplete knowledge bases yield non-exhaustive ground truth, contaminating correctness measurement.
Multi-answer (controlled)	**MACE (Ours)**	✓	✓	✓	—

Table 5: **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 questions retained 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

of answer multiplicity from missing labels or unmatched question designs. **MACE** is constructed to satisfy **P1–P3** jointly.

B The MACE Benchmark

This appendix details the three construction phases summarized in §3.2 and the evidence that MACE satisfies P1–P3 (§3.1): domain selection (§B.1) and triplet collection (§B.2) for knowledge collection, heuristic filtering (§B.3) and manual verification (§B.4) for knowledge filtering, and the templates and constraint-relaxation procedure that control cardinality (§B.5).

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.

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.

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 exploration of the domain space, it introduces 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 skew toward domains 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 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 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 number of valid answers per instance.

Except for the *Math* domain, which is generated through rule-based synthesis, all other domains are collected via the Wikidata SPARQL endpoint[2]. For each domain, we query Wikidata to retrieve relevant subjects and their associated factual objects based on predefined relations (see Table 7). 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

Table 6: **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

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, introducing noise into the evaluation. We therefore rank all entity labels within each domain by their 12-month aggregate Wikipedia pageview counts, a proxy for public familiarity, and apply two complementary retention criteria: (i) we retain any item whose pageview count exceeds 1,000, ensuring that all entities with meaningful public recognition are preserved regardless of intra-domain rank; (ii) within each domain, we 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 review. Together these criteria balance broad entity coverage against practical annotation cost, ensuring that 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 pointing 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 7).

Domain-Specific Refinement Criteria. Beyond these universal checks, we impose domain-specific refinement criteria. The 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 penalized simply because it is absent from an incomplete ground-truth set. Since exhaustive ground truth (P3) is a hard requirement for our analysis, the rules below serve as heuristic proxies for *completeness*: by retaining only entities whose Wikidata records show signs of thorough curation (e.g. continuous yearly entries, sufficient object counts), we maximize the likelihood that 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

Table 7: **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 ($s \to o$)	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

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 8), 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 is a deliberate choice driven by our evaluation goal: attributing changes in confidence reliability to the number of valid answers requires holding the remaining sources of variation fixed.

Concretely, the controlled template structure offers three methodological advantages:

Table 8: **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.

- **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. A change in confidence behavior between the 1-answer and 6-answer settings can therefore be attributed to answer multiplicity 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. For a benchmark whose purpose is to isolate the effect of answer cardinality under controlled difficulty, however, 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 future work.

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 8, 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.

Table 9: **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	$T_n = \frac{n(n+1)}{2},\ n \in \mathbb{N}$

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 9), 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 a chosen number type, the reference set contains all integers within the specified interval that satisfy its defining predicate. We set the interval `{lower_num}`–`{upper_num}` so that this set contains exactly k answers. This provides wide coverage of numerical reasoning behaviors while maintaining precise answer-count control. Given the virtually unlimited pool of candidate integers, we cap the total number of generated questions at 10,000.

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 normalized 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

This appendix gives the formulations and prompts of the 15 estimators (§C.1), the inference and aggregation settings (§C.2), and why all metrics are macro-averaged over the six domains (§C.3). Estimators appear in the four groups of §4 and under the names used in the main text.

C.1 CONFIDENCE ESTIMATION METHODS

Definitions below follow the four groups of §4, in the same order and under the same names used throughout the paper.

(i) The whole predictive distribution.

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

$$U_{\text{Prob Entropy}} = -\frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{|r_j|}\sum_{v\in\mathcal{V}} p_{j,i}(v)\log p_{j,i}(v), \tag{2}$$

  where p_{z_i} represents the normalized probability distribution over tokens within the j-th generation.
- **N-Prob Entropy** (Malinin & Gales, 2020) To avoid bias toward longer sequences, we normalize the entropy of each sampled response by its length:

$$U_{\text{N-Prob Entropy}} = -\frac{1}{M}\sum_{j=1}^{M}\frac{1}{|r_j|}\sum_{i=1}^{|r_j|}\sum_{v\in\mathcal{V}} p_{j,i}(v)\log p_{j,i}(v). \tag{3}$$

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

$$P(C_k) = \sum_{r_j\in C_k} P(r_j), \tag{4}$$

  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:

$$\tilde{P}(C_k) = \frac{P(C_k)}{\sum_{i=1}^{K} P(C_i)}, \tag{5}$$

  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:

$$U_{\text{SE}} = -\sum_{k=1}^{K}\tilde{P}(C_k)\log\tilde{P}(C_k), \tag{6}$$

  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. Note that this reading of dispersion as uncertainty is precisely the assumption our study examines when several answers are valid.

(ii) The support of the returned answer.

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

$$U_{\text{Perplexity}} = -\frac{1}{|r|}\sum_{i=1}^{|r|}\log p_{z_i}, \tag{7}$$

  where r denotes the generated sequence and p_{z_i} is the model-assigned probability for the i-th token. The corresponding confidence is $\exp(-U_{\text{Perplexity}})$, the geometric mean of the token probabilities. Length normalization matters here because MACE answers are short entities whose token counts differ across domains, so the un-normalized sequence probability would confound answer length with correctness.
- **Consis** (Manakul et al., 2023) The consistency score measures the proportion of model samples that agree with the greedy returned answer $\hat{y}_0$ among M sampled outputs:

$$C_{\text{consistency}} = \frac{1}{M}\sum_{i=1}^{M}\mathbb{I}\{\hat{Y}_i = \hat{y}_0\}, \tag{8}$$

  where $\mathbb{I}\{\cdot\}$ equals 1 if the i-th sampled answer matches the greedy returned answer, and 0 otherwise.

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}
```

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

$$C_{\text{conf}} = \frac{\sum_{i=1}^{M} \mathbb{I}\{\hat{Y}_i = \hat{y}_0\} \times C_i}{\sum_{i=1}^{M} C_i}, \tag{9}$$

  where agreement is measured against the greedy returned answer and higher-confidence samples contribute proportionally more to the final score. When C_i is obtained from the *Verb* or *Verb-Topk* methods, this variant corresponds to *Consis-Verb* and *Consis-Verb-Topk*, respectively.

Both entropies and Perplexity are measures of *uncertainty*, and we convert them to confidences as follows.

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:

$$\text{Conf}(x) \ = \ e^{-H(x)}, \tag{10}$$

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. Because the transformation is strictly monotonic, it preserves instance ordering and therefore leaves AUROC unchanged, while placing all estimators on a common scale for the mean-confidence comparisons in Table 1.

(iii) The model's self-report about the returned answer.

- **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:

$$C_{\text{Verb}} = \text{LLM}(x). \tag{11}$$

- **Verb-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

Prompt Template for Verb-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.
```

$\{C_1, C_2, \ldots, C_k\}$. The implementation selects the candidate with the largest stated confidence and uses that candidate and its confidence as the returned answer and score:

$$j_0 = \arg\max_{1 \le i \le k} C_i, \qquad C_{\text{Verb-Topk}} = C_{j_0}, \qquad \hat{y}_0 = \hat{y}_{j_0}. \tag{12}$$

Correctness is then evaluated separately against the verified answer set.

(iv) A separate verification query.

- **P_{True}-Consis / P_{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`":

$$C_{\text{P}_{\text{True}}} = p(z_{\text{true}} \mid x'). \tag{13}$$

  We implement two variants: (1) In the **P_{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_{True}-Prob** variant, we directly use the model's internal probability (logit-derived softmax value) assigned to the "`True`" token, without repeated sampling. For P_{True}-Consis, the implementation uses $T = 1$ and 20 independent verdict samples.
- **P_{True}-Consis-Cand / P_{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:

$$C_{\text{P}_{\text{True}}\text{-Consis-Cand}} = p(z_{\text{true}} \mid x', \{a_1, a_2, \ldots, a_K\}). \tag{14}$$

  Two variants are considered: (1) In the **P_{True}-Consis-Cand** variant, confidence is computed as the empirical frequency of "`True`" responses across multiple sampling runs; (2) In the **P_{True}-Prob-Cand** variant, the probability assigned to the "`True`" token from the model's logits is used directly as the confidence score.
- **Self-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") reflecting how certain it is about $\hat{y}$. The resulting confidence is defined as:

$$C_{\text{Self-Ask}} = \text{LLM}(x', \hat{y}), \tag{15}$$

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
```

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

- **Self-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:

$$C_{\text{Self-Ask-Cand}} = \text{LLM}(x', \hat{y}, \{a_1, a_2, \ldots, a_K\}), \tag{16}$$

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

C.2 Experiment Details

Inference Framework. Open-weight models are run with **vLLM** (Kwon et al., 2023) for batched decoding and token-level log-probabilities. API-based models (GPT-4o, GPT-4o-mini, and DeepSeek-V3) are queried through their respective APIs; methods requiring internal token probabilities are

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-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}
```

omitted, except that Sem Entropy uses the empirical sample-frequency cluster distribution described above.

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

- **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, and all distribution-side quantities in §5.3): 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.

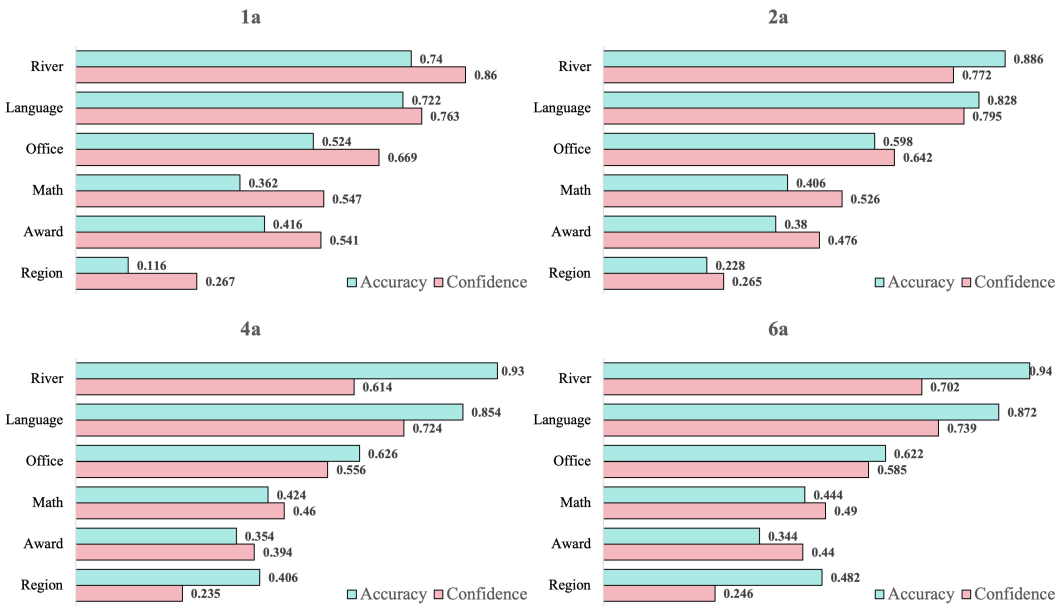

Figure 4: Domain-specific accuracy and confidence under different ground-truth sizes.

Entropy-to-Confidence Conversion. For all entropy-based estimators (Prob Entropy, N-Prob Entropy and Sem Entropy), the raw entropy value H is converted into a confidence score via the exponential transformation $\mathrm{Conf} = e^{-H}$, as described in Eq. (10). This places all methods on a common $(0,\ 1]$ scale without changing their instance ordering.

LLM-based Evaluation. We use `GPT-4o` as the automated evaluator for two critical components of our pipeline, following Liu et al. (2023):

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. An answer is deemed correct if it matches *any* ground-truth answer.
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, since 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 procedure 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.

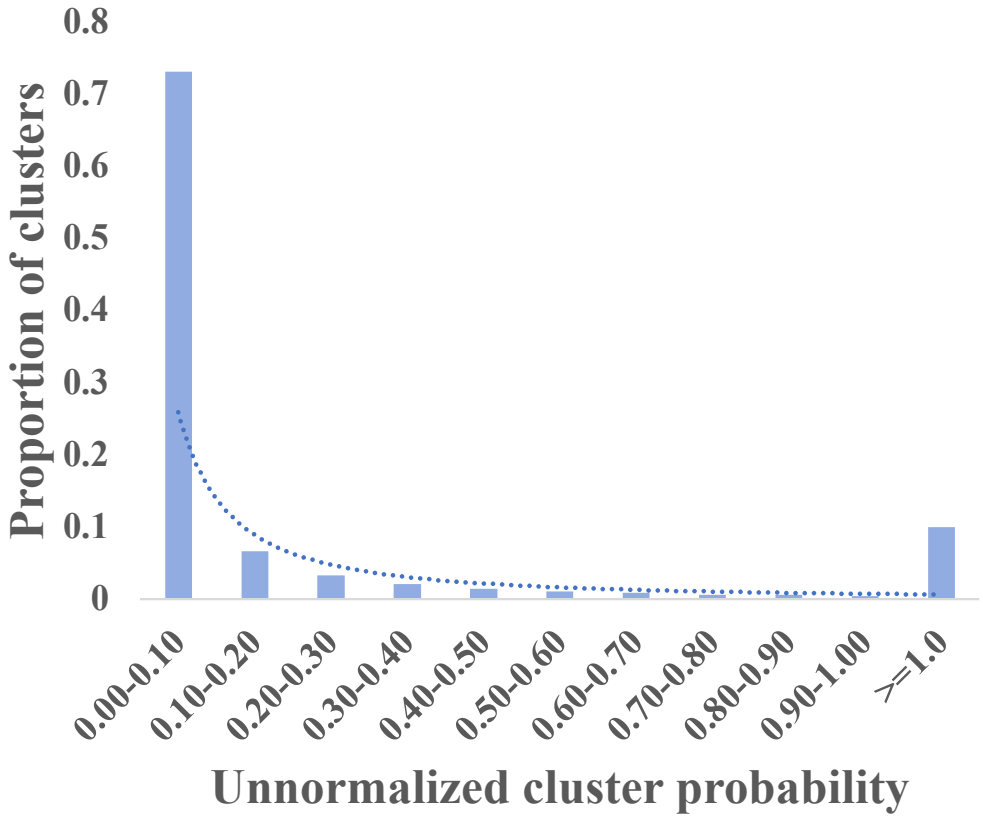

Figure 5: **Distribution of unnormalized cluster probabilities** $P_{\text{raw}}(C_m) = \sum_{r_j \in C_m} P(r_j)$ for **LLaMA-3.1-70B-Instruct**. Each bar shows the fraction of semantic clusters whose cumulative probability falls within the corresponding bin. Since probabilities are summed over $M = 20$ samples without renormalization, values can exceed 1. Over 70% of clusters concentrate in $[0,\ 0.1)$, motivating our filtering threshold $\tau = 0.1$ (see §D.11). The dotted curve is a kernel density estimate for visual reference.

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 the specified mixture when reporting mixed-cardinality AUROC—and compute one AUROC value per domain on this pooled set. The final reported number is the unweighted mean of the six domain-level AUROC scores.
3. **Inter-domain averaging.** In both cases 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 behavior of **LLaMA-3.1-70B-Instruct** across varying ground-truth set sizes. As illustrated in Figure 4, all configurations (`1a/2a/4a/6a`) exhibit a consistent trend: domains with higher accuracy (lower intrinsic difficulty) also show 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*. A *micro-average* (pooling all instances regardless of domain) would be dominated by the high-accuracy, low-difficulty domains, masking weak confidence reliability in harder domains

where it 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 scheme throughout the paper. This treats every domain as an equally weighted evaluation scenario and gives a more faithful picture of global reliability, especially for safety-critical applications where worst-case domain behavior is at least as important as best-case behavior.

Structural Origins of Difficulty Variation. Importantly, 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 8; Appendix §B.5). It originates instead from the two 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 a *large candidate pool* of entities sharing 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 ($k \in \{1, 2, 4, 6\}$), which is what allows us to read differences across settings as effects of answer cardinality.

D Results and Analysis

This appendix supports §5 and §6 in the order the main text presents them. For the inversion and the redistribution of valid mass (§5.1), §D.1 reports every model and §D.2 states how the distributional quantities are measured. For the loss of discrimination under mixed cardinalities (§5.2), §D.3 separates the within-cardinality loss, §D.4 prices it in selective prediction, §D.5 rules out a base-rate artifact, §D.6 repeats the mixture ladder on five further models, and §D.7 explains why we do not report calibration error. For the mechanism (§5.3), §D.8 gives the per-estimator evidence, §D.9 derives the entropy decomposition, and §D.10 relates a model's sensitivity to its valid-answer coverage. §D.11 documents the semantic grouping all distributional quantities rest on, and §D.12 supports CARE (§6).

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`) in Tables 10–17; LLaMA-3.1-70B-Instruct, the running example, appears as Table 1 in the main text. For API-based models (GPT-4o, GPT-4o-mini, and DeepSeek-V3), we omit methods that require logit probabilities or internal token distributions, such as Prob Entropy, N-Prob Entropy, and Perplexity. For entropy-based methods we convert entropy H into a confidence score via $\text{conf} = e^{-H}$, so that higher values uniformly indicate greater confidence.

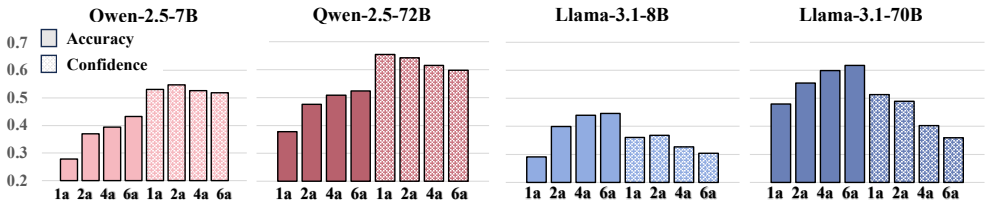

Figure 6: **The multi-answer confidence inversion across model scales.** Accuracy and consistency-based confidence for a small and a large model in each of two families, over the four answer cardinalities. Accuracy rises with cardinality while confidence declines, and the two separate further for the larger model of each pair, the observation §5.3 attributes to broader valid-answer coverage.

Table 10: **Average confidence score under each answer cardinality** for Qwen2.5-7B-Instruct. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend.

Method	**1a**	**2a**	**4a**	**6a**	Δ
(i) Whole predictive distribution					
Prob Entropy$^{e^{-H}}$	67.8	68.2	66.9	66.6	-1.2
N-Prob Entropy$^{e^{-H}}$	92.1	92.3	92.0	91.9	-0.2
Sem Entropy$^{e^{-H}}$	64.7	65.5	63.6	62.9	-1.8
(ii) Support of the returned answer					
Perplexity$^{e^{-H}}$	79.2	79.6	79.4	79.1	-0.1
Consis	53.1	54.7	52.6	51.9	-1.2
Consis-Verb	44.9	44.3	41.2	40.5	-4.4
Consis-Verb-Topk	35.1	35.3	31.1	30.7	-4.4
(iii) Self-report on the returned answer					
Verb	94.7	95.4	94.8	94.7	0.0
Verb-Topk	82.5	81.3	73.7	71.8	-10.7
(iv) Separate verification query					
P_{True}-Consis	46.9	47.0	45.0	46.9	0.0
P_{True}-Prob	46.8	46.9	44.9	46.8	0.0
P_{True}-Consis-Cand	59.3	61.0	62.9	65.0	$+5.7$
P_{True}-Prob-Cand	59.4	61.1	63.0	65.2	$+5.8$
Self-Ask	73.7	73.8	73.0	73.7	0.0
Self-Ask-Cand	70.5	71.5	71.1	72.2	$+1.7$

Across model families and parameter scales, one pattern recurs: accuracy trends steadily upward as the number of valid answers increases, while confidence declines over the same range for the scores that read the generation distribution or the model's self-report. The exceptions are the candidate-aware verification variants on the smaller models, whose scores are flat or slightly increasing; §D.8 attributes this to their judging each candidate on its own rather than against competing answers. The two quantities that ought to move together therefore move apart, and as more answers become acceptable the models appear increasingly *underconfident*. How the quantities in these tables are computed from the samples is stated in §D.2, and the consequence for correctness ranking is quantified in §5.2 and expanded in §D.3–§D.7.

D.2 Instruments Computed from the Verified Answer Sets

Several quantities in §5.1 and §5.3 are computed against the verified answer sets and are therefore instruments available to us and not to an estimator. This section states how they are obtained and what limits their interpretation.

All of them are read off the same $N = 20$ sampled responses that the estimators consume. For a question x with valid-answer set $\mathcal{A}(x)$, the published accuracy labels use the GPT-4o correctness verdict described in §C.2. The *valid mass* $q(x)$ is obtained by clustering the samples into semantic

Table 11: **Average confidence score under each answer cardinality** for Qwen2.5-14B-Instruct. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Prob Entropy$^{e^{-H}}$	73.4	73.3	72.4	71.7	-1.7
N-Prob Entropy$^{e^{-H}}$	93.7	93.6	93.5	93.2	-0.5
Sem Entropy$^{e^{-H}}$	70.6	71.3	69.9	68.4	-2.2
(ii) Support of the returned answer					
Perplexity$^{e^{-H}}$	85.6	85.8	85.8	85.4	-0.2
Consis	62.7	63.3	62.5	61.0	-1.7
Consis-Verb	42.2	45.6	43.4	41.7	-0.5
Consis-Verb-Topk	45.5	44.8	42.7	40.7	-4.8
(iii) Self-report on the returned answer					
Verb	87.4	87.1	85.6	85.4	-2.0
Verb-Topk	87.6	84.0	74.1	70.5	-17.1
(iv) Separate verification query					
P_{True}-Consis	50.1	52.2	49.4	49.5	-0.6
P_{True}-Prob	50.1	52.2	49.4	49.5	-0.6
P_{True}-Consis-Cand	67.1	69.8	71.8	71.9	$+4.8$
P_{True}-Prob-Cand	67.1	69.8	71.8	71.9	$+4.8$
Self-Ask	77.5	78.7	76.0	75.2	-2.3
Self-Ask-Cand	76.2	78.2	78.0	77.4	$+1.2$

Table 12: **Average confidence score under each answer cardinality** for Qwen2.5-32B-Instruct. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Prob Entropy$^{e^{-H}}$	74.9	74.2	73.4	72.7	-2.2
N-Prob Entropy$^{e^{-H}}$	94.1	94.1	93.8	93.6	-0.5
Sem Entropy$^{e^{-H}}$	72.8	72.4	71.2	69.1	-3.7
(ii) Support of the returned answer					
Perplexity$^{e^{-H}}$	85.9	85.7	85.5	85.0	-0.9
Consis	64.9	63.9	62.9	61.2	-3.7
Consis-Verb	54.1	52.7	51.8	50.5	-3.6
Consis-Verb-Topk	48.1	45.9	44.7	42.2	-5.9
(iii) Self-report on the returned answer					
Verb	88.6	88.1	87.3	87.7	-0.9
Verb-Topk	87.6	82.4	75.1	72.4	-15.2
(iv) Separate verification query					
P_{True}-Consis	69.4	76.2	78.2	78.4	$+9.0$
P_{True}-Prob	69.5	76.2	78.3	78.5	$+9.0$
P_{True}-Consis-Cand	74.9	78.7	81.0	81.9	$+7.0$
P_{True}-Prob-Cand	74.9	78.7	81.1	81.9	$+7.0$
Self-Ask	72.8	72.5	70.7	72.4	-0.4
Self-Ask-Cand	73.2	75.2	75.6	76.1	$+2.9$

modes (§D.11), labeling a mode valid when the majority of its members are valid, and summing the normalized cluster probabilities of the valid modes; $M^+(x)$ counts the distinct valid modes the samples cover. Restricting to questions the model answers correctly, the *share* $r_1(x)$ is the returned answer's mode probability divided by $q(x)$, so that qr_1 is the probability on the returned answer and $q(1 - r_1)$ the probability on the model's other valid answers.

Two caveats bound what these quantities establish. First, they inherit error in the automated correctness judgments. Because MACE answer sets are exhaustive and expert-verified (Cohen's $\kappa = 0.94$),

Table 13: **Average confidence score under each answer cardinality** for Qwen2.5-72B-Instruct. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Prob Entropy$^{e^{-H}}$	76.4	75.4	74.3	73.4	-3.0
N-Prob Entropy$^{e^{-H}}$	94.1	93.9	93.5	93.3	-0.8
Sem Entropy$^{e^{-H}}$	71.8	70.3	68.3	66.6	-5.2
(ii) Support of the returned answer					
Perplexity$^{e^{-H}}$	87.7	87.2	86.2	85.5	-2.2
Consis	65.4	64.2	61.5	59.8	-5.6
Consis-Verb	59.0	58.2	55.5	53.7	-5.3
Consis-Verb-Topk	53.2	52.8	48.1	45.0	-8.2
(iii) Self-report on the returned answer					
Verb	87.9	87.3	86.2	86.8	-1.1
Verb-Topk	85.9	79.5	70.8	69.9	-16.0
(iv) Separate verification query					
P_{True}-Consis	75.1	73.3	68.6	73.7	-1.4
P_{True}-Prob	75.2	73.3	68.8	73.9	-1.3
P_{True}-Consis-Cand	75.8	76.0	73.4	77.6	$+1.8$
P_{True}-Prob-Cand	75.8	76.1	73.5	77.7	$+1.9$
Self-Ask	84.9	79.7	72.9	75.3	-9.6
Self-Ask-Cand	83.5	81.8	76.6	79.5	-4.0

Table 14: **Average confidence score under each answer cardinality** for LLaMA-3.1-8B-Instruct. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Prob Entropy$^{e^{-H}}$	63.4	62.5	60.6	59.5	-3.9
N-Prob Entropy$^{e^{-H}}$	89.7	89.3	88.8	88.6	-1.1
Sem Entropy$^{e^{-H}}$	49.9	51.4	47.5	45.1	-4.8
(ii) Support of the returned answer					
Perplexity$^{e^{-H}}$	67.0	67.6	65.9	64.4	-2.6
Consis	36.0	36.8	32.7	30.4	-5.6
Consis-Verb	37.1	37.0	33.5	30.7	-6.4
Consis-Verb-Topk	30.4	29.8	26.9	24.6	-5.8
(iii) Self-report on the returned answer					
Verb	96.2	96.1	96.2	96.2	0.0
Verb-Topk	95.8	96.3	96.2	96.3	$+0.5$
(iv) Separate verification query					
P_{True}-Consis	53.3	50.8	46.2	45.9	-7.4
P_{True}-Prob	53.1	50.9	46.4	45.7	-7.4
P_{True}-Consis-Cand	64.8	65.7	66.1	65.7	$+0.9$
P_{True}-Prob-Cand	64.7	65.7	65.9	65.9	$+1.2$
Self-Ask	37.7	31.7	26.8	27.2	-10.5
Self-Ask-Cand	45.4	47.8	46.3	46.1	$+0.7$

we expect this error to be small, but it means the values locate the quantities they estimate only approximately. The standalone decoupling reproduction script uses a local exact/substring/fuzzy matcher and is intended as an approximate audit of the published values. Second, the samples cover the distribution only partially: the distinct answers they produce account for 62–67% of the total sequence probability, so q and M^+ are truncated views, and mass on answers never sampled is invisible to the instruments exactly as it is to the estimators. Both caveats affect the absolute level of these quantities more than their trend across cardinality, which is what §5.3 reads off them.

Table 15: **Average confidence score under each answer cardinality** for GPT-4o-mini. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend. Scores requiring token probabilities are unavailable for this API-based model.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Sem Entropy$^{e^{-H}}$	49.6	50.1	46.2	44.9	-4.7
(ii) Support of the returned answer					
Consis	60.1	60.3	57.6	56.5	-3.6
Consis-Verb	54.1	54.3	51.2	50.2	-3.9
Consis-Verb-Topk	48.7	46.3	42.4	41.1	-7.6
(iii) Self-report on the returned answer					
Verb	93.2	92.1	91.1	90.8	-2.4
Verb-Topk	83.7	73.0	61.5	56.5	-27.2
(iv) Separate verification query					
P_{True}-Consis	62.5	67.7	68.3	70.5	$+8.0$
P_{True}-Consis-Cand	72.3	77.0	77.6	79.1	$+6.8$
Self-Ask	76.3	76.1	73.7	74.9	-1.4
Self-Ask-Cand	73.7	76.4	76.9	76.6	$+2.9$

Table 16: **Average confidence score under each answer cardinality** for GPT-4o. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend. Scores requiring token probabilities are unavailable for this API-based model.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Sem Entropy$^{e^{-H}}$	63.0	54.1	45.4	40.4	-22.6
(ii) Support of the returned answer					
Consis	70.1	64.3	56.5	52.6	-17.5
Consis-Verb	68.8	62.8	54.8	50.6	-18.2
Consis-Verb-Topk	65.1	58.5	48.8	45.8	-19.3
(iii) Self-report on the returned answer					
Verb	92.1	90.7	88.8	88.5	-3.6
Verb-Topk	89.6	85.6	71.8	67.4	-22.2
(iv) Separate verification query					
P_{True}-Consis	64.3	66.8	68.4	70.4	$+6.1$
P_{True}-Consis-Cand	80.0	81.0	80.5	80.0	0.0
Self-Ask	77.8	75.8	74.9	75.3	-2.5
Self-Ask-Cand	80.5	80.4	79.0	77.9	-2.6

D.3 Discrimination Within a Single Cardinality

§5.2 pools cardinalities, which is the regime deployment presents, and attributes the loss to comparisons *across* cardinalities. Scoring each cardinality separately isolates the remaining half of the effect. Table 18 reports AUROC computed inside each cardinality, so no cross-cardinality comparison enters the metric.

Discrimination degrades there too, and by more than pooling costs: Consis falls from 80.1 on one-answer questions to 70.1 on six-answer questions, against 5.6 points for the pooled mixture, and Verb-Topk from 60.0 to 48.9. The two losses are distinct. The within-cardinality loss says that a score discriminates less well among questions that all admit six answers, because r_1 varies more across those questions; the pooled loss adds the misalignment of score levels between cardinalities, and rank-aligning within cardinality recovers between a quarter and 45% of it (Appendix §D.7).

The table also shows why the magnitude of a confidence shift says nothing about either loss. N-Prob Entropy moves 1.8 confidence points from 1a to 6a and loses 19.0 AUROC within cardinality; Verb-Topk moves 25.0 points for a smaller 11.1 within cardinality. The two are not on a common scale, since the e^{-H} map of §4 fixes the range of one and not the other, and neither shift is a monotone function of what is lost.

Table 17: **Average confidence score under each answer cardinality** for DeepSeek-V3. "1a" denotes questions with one valid answer and Δ the change from 1a to 6a. Groups and row order follow §4, as in Table 1. Scores marked e^{-H} are uncertainties mapped to confidence to display the trend. Scores requiring token probabilities are unavailable for this API-based model.

Method	1a	2a	4a	6a	Δ
(i) Whole predictive distribution					
Sem Entropy$^{e^{-H}}$	80.9	78.6	76.2	72.9	-8.0
(ii) Support of the returned answer					
Consis	81.1	79.1	76.7	73.2	-7.9
Consis-Verb	69.0	65.9	60.6	55.6	-13.4
Consis-Verb-Topk	63.4	58.5	53.6	49.0	-14.4
(iii) Self-report on the returned answer					
Verb	85.3	85.5	85.8	86.0	$+0.7$
Verb-Topk	63.5	57.5	52.7	50.5	-13.0
(iv) Separate verification query					
P_{True}-Consis	64.3	64.6	64.7	66.2	$+1.9$
P_{True}-Consis-Cand	82.4	81.5	82.5	80.5	-1.9
Self-Ask	76.5	72.8	69.8	71.2	-5.3
Self-Ask-Cand	83.8	81.3	80.8	78.7	-5.1

Table 18: **AUROC computed within each answer cardinality separately** for LLaMA-3.1-70B-Instruct. Column "6a" scores only questions with six valid answers, so no cross-cardinality comparability is required. Δ is 1a to 6a; the last column repeats the pooled loss of Table 3 for contrast.

Method	1a	2a	4a	6a	Δ	Δpooled
(i) Whole predictive distribution						
Prob Entropy	73.8	71.9	66.5	58.3	-15.5	-8.5
N-Prob Entropy	72.6	67.5	59.2	53.6	-19.0	-10.1
Sem Entropy	78.9	79.0	75.2	66.7	-12.2	-6.8
(ii) Support of the returned answer						
Consis	80.1	80.3	77.3	70.1	-10.1	-5.6
Perplexity	80.1	78.2	73.0	67.3	-12.8	-7.1
Consis-Verb	79.1	79.2	75.1	68.0	-11.1	-5.8
Consis-Verb-Topk	79.7	78.0	76.5	68.3	-11.4	-5.8
(iii) Self-report on the returned answer						
Verb	58.0	59.6	58.1	54.5	-3.5	-2.5
Verb-Topk	60.0	58.0	52.5	48.9	-11.1	-11.6
(iv) Separate verification query						
P_{True}-Consis	68.4	66.7	63.5	65.5	-2.9	-5.2
P_{True}-Prob	72.5	70.4	66.5	67.6	-4.9	-6.3
P_{True}-Consis-Cand	62.5	62.9	63.7	62.0	-0.5	-0.1
P_{True}-Prob-Cand	75.1	73.1	72.9	69.5	-5.6	-4.5
Self-Ask	65.6	65.5	62.1	61.7	-4.0	-2.7
Self-Ask-Cand	65.9	67.1	68.2	64.8	-1.1	-0.3

D.4 Selective Prediction

AUROC summarizes ranking quality, and the use it stands in for is selective prediction: answer the queries whose score clears a threshold, defer the rest. One threshold has to serve the whole stream, which is exactly the condition §5.2 studies, so we report what the ranking is worth at fixed coverage.

All numbers are macro-averaged over domains, as elsewhere in the paper, with the Math domain averaged over its five number types.

Table 19 makes the accounting explicit because the raw number is misleading on its own. Answering the highest-scoring half of the pool yields 68.1% accuracy at cardinality 1 and 70.1% in the broadest mixture for *Consis*, which appears to improve. It does not: answering *everything* yields 48.0% and 56.3% over the same range, so the accuracy the score adds falls from $+20.1$ to $+13.8$ points. This is

Table 19: **Selective prediction under cumulative cardinality mixtures** (LLaMA-3.1-70B-Instruct). *Gain* is the accuracy on the answered subset minus the accuracy of answering every question in the same pool, which is 48.0, 51.8, 54.5 and 56.3 for the four mixtures. Δ is [1] minus $[1, 2, 4, 6]$.

Method	**[1]**	**[1,2]**	**[1,2,4]**	**[1,2,4,6]**	Δ
Accuracy on the answered subset, 50% *coverage*					
Consis	68.1	69.4	70.4	70.1	+2.0
Perplexity	67.7	69.1	69.6	69.7	+2.0
Gain over answering everything, 50% *coverage*					
Consis	20.1	17.6	15.9	13.8	−6.3
Perplexity	19.7	17.4	15.1	13.4	−6.3
Gain over answering everything, 20% *coverage*					
Consis	27.2	26.2	24.6	21.9	−5.3
Perplexity	30.2	28.0	25.7	23.9	−6.3

the same cancellation between rising accuracy and falling scores that misleads ECE (§D.7), and the gain over the all-coverage baseline is the quantity that isolates the score's contribution.

The loss of gain is consistent in size and largely indifferent to the operating point: about six points at 50% coverage and five to six at 20% coverage. Restricting to the most confident fifth therefore does not recover what pooling costs, and the two strongest estimators lose comparable amounts, so the degradation is a property of the setting rather than of a particular score.

D.5 Pooled Degradation Is Not a Base-Rate Artifact

Because accuracy rises with cardinality while scores fall (§5.1), part of the pooled decline reported in §5.2 could in principle come from a shift in class balance rather than from the scores. This section rules that out in two ways. Pooled AUROC is a weighted average of comparisons, one for each pair of cardinalities the correct and the incorrect answer can be drawn from, so it can be decomposed exactly. Table 20 gives that decomposition. The diagonal reproduces the single-cardinality AUROCs, and the off-diagonal cells carry the failure: for every score the pair (correct answer at 6a, incorrect answer at 1a) is the worst cell and its mirror is the best, which is the confidence inversion of §5.1 expressed in the metric the paper reports.

This decomposition also locates where the base rate acts. AUROC is insensitive to class balance within a fixed population, so the base rate cannot move the individual cells; what it moves are the weights. Accuracy rises with cardinality, so the pooled positives are drawn disproportionately from the high cardinalities, which are exactly the ones that lose the cross-cardinality comparison. Holding every cell fixed and replacing the observed class shares with accuracy-matched ones moves pooled AUROC by 1.1 points for *Consis*, so reweighting accounts for part of the decline but not for most of it.

We then verify this by intervention rather than by decomposition. Within each domain and cardinality we discard correct predictions at random until every cardinality has the accuracy of the hardest one (0.475, the value at cardinality 1), which equalizes the base rate while leaving each score untouched, and recompute the mixture ladder. Results are averaged over five random subsamples.

Every estimator still degrades, and the ordering across families is unchanged (Table 21). Matching the base rate removes about half of the magnitude and none of the effect: each score retains 46 to 59% of its uncontrolled degradation. The residual is what the mechanism of §5.3 predicts should survive, since re-dividing valid mass lowers a multi-answer question's score whether or not its accuracy is held down artificially.

D.6 AUROC under Cardinality Mixtures Across Models

Tables 22–26 repeat the mixture ladder of §5.2 on five further models: Qwen2.5-Instruct at 7B, 14B, 32B and 72B, and LLaMA-3.1-8B-Instruct. The pattern of §5.2 holds on all of them. Pooling cardinalities costs 3.8 to 5.3 AUROC points on average across the 15 estimators, and all 75 model-estimator pairs decline without exception.

Table 20: **Cross-cardinality decomposition of pooled AUROC** (LLaMA-3.1-70B-Instruct, domain-macro, in points). Each entry is the probability that a correct answer to a question of the row cardinality receives a higher score than an incorrect answer to a question of the column cardinality. Underlined diagonal entries are the single-cardinality AUROCs of Appendix §D.3; the minimum of each block is in bold. The pattern holds for every score: the (6a correct, 1a incorrect) cell is the minimum, entries rise along a row as the incorrect answer is drawn from a higher cardinality, and the 6a row sits below the 1a row throughout.

	Incorrect answer drawn from			
Correct answer from	**1a**	**2a**	**4a**	**6a**
Consis				
1a	<u>80.1</u>	83.2	86.8	87.2
6a	**59.5**	63.8	69.9	<u>70.1</u>
Perplexity				
1a	<u>80.1</u>	82.5	84.7	85.5
6a	**59.4**	62.4	65.8	<u>67.3</u>

Table 21: **AUROC under cumulative mixtures after equalizing accuracy across cardinalities** (LLaMA-3.1-70B-Instruct, mean of five subsamples). Δ is $[1]$ minus $[1, 2, 4, 6]$; the last column repeats the uncontrolled value from Table 3 for comparison.

Method	**[1]**	**[1,2]**	**[1,2,4]**	**[1,2,4,6]**	Δ	Δ **uncontr.**
Sem Entropy	78.9	78.8	77.5	75.4	-3.4	-6.8
Consis	80.1	80.1	79.3	77.6	-2.6	-5.6
Perplexity	80.1	79.4	77.9	75.9	-4.2	-7.1

The ordering across groups reproduces §5.2: scores over the whole predictive distribution and over the returned answer's support give up the most, separate verification queries the least, and self-report spans the widest range within a single group. *Verb-Topk* is the largest single loss on three of the five models (9.2 to 10.0 points), falling to near-chance discrimination in the broadest mixture, while plain *Verb* moves little from an already weak starting point. Which estimator is strongest in absolute terms does vary with the model—*Perplexity* on the two smaller Qwen models, P_{True}-*Prob* on Qwen2.5-32B, *Prob Entropy* on Qwen2.5-72B and *Consis* on LLaMA-3.1-8B—but no model supplies an estimator that is stable across cardinalities, which is the claim these tables are reported to test.

Each table also carries the **CARE** row of §6, so what the answer count is worth can be read per model rather than only on the representative one. Supplied with the verified count, CARE gains 2.1 to 3.2 points over *Consis* in the broadest mixture on these five models and reduces the pooled degradation on every one, from a range of -2.5 to -4.1 down to $+0.2$ to -1.7. On the two largest models here it removes the degradation outright. The quantity the correction needs is therefore not specific to the model whose mechanism §5.3 measures.

D.7 Expected Calibration Error

What a calibration number presumes. ECE compares a score against the frequency with which answers at that score are correct, so it presumes the score is an estimate of that frequency. On this point the estimators fall into three groups, and only the first admits a calibration number directly. **(a) Scores that are estimates of correctness.** *Verb*, *Verb-Topk*, *Self-Ask* and the four P_{True} variants are asked for the probability that the produced answer is correct and return a number on that scale. ECE reads them as they were meant. **(b) Scores that estimate the probability of a different event.** *Perplexity* is the per-token geometric mean of the generation probability, *Consis* is the sampled frequency of the returned mode, and the *Consis-Verb* hybrids are that frequency with verbalized weights, so each estimates how much mass the model places on the answer it returned. Reading either as a probability of correctness presumes the single-answer identification of §5.1, and their ECE therefore measures how well that identification holds—which is the quantity multiple valid answers destroy, not a property of the estimator.

Table 22: **AUROC on Qwen2.5-7B-Instruct under cumulative answer-cardinality mixtures.**

Method	≤ 1	≤ 2	≤ 4	≤ 6	Δ
(i) Whole predictive distribution					
Prob Entropy	75.6	75.4	74.2	72.8	-2.8
N-Prob Entropy	75.2	73.9	72.6	71.8	-3.4
Sem Entropy	74.0	74.1	72.7	71.2	-2.8
(ii) Support of the returned answer					
Consis	76.7	**76.5**	**75.4**	74.2	-2.5
Perplexity	**77.1**	76.4	75.0	**74.3**	-2.8
Consis-Verb	74.6	73.7	73.3	72.5	-2.1
Consis-Verb-Topk	75.9	73.5	72.5	71.7	-4.2
(iii) Self-report on the returned answer					
Verb	57.5	56.2	55.9	54.6	-2.9
Verb-Topk	61.4	57.2	53.6	52.2	-9.2
(iv) Separate verification query					
P_{True}-Consis	70.1	66.1	66.7	66.3	-3.8
P_{True}-Prob	75.6	70.6	70.7	69.8	-5.8
P_{True}-Consis-Cand	67.1	68.0	67.3	66.3	-0.8
P_{True}-Prob-Cand	75.6	75.1	73.7	72.4	-3.2
Self-Ask	69.1	65.3	64.2	62.2	-6.9
Self-Ask-Cand	74.0	73.1	71.5	70.2	-3.8
Cardinality-rescaled (§6, verified count)					
CARE	76.7	77.5	77.4	76.9	$+0.2$

Table 23: **AUROC on Qwen2.5-14B-Instruct under cumulative answer-cardinality mixtures.**

Method	≤ 1	≤ 2	≤ 4	≤ 6	Δ
(i) Whole predictive distribution					
Prob Entropy	79.8	77.6	75.5	74.2	-5.6
N-Prob Entropy	80.4	78.3	76.1	74.4	-6.0
Sem Entropy	77.4	77.2	74.8	73.4	-4.0
(ii) Support of the returned answer					
Consis	78.1	77.5	75.5	74.3	-3.8
Perplexity	**81.5**	**79.6**	**76.9**	**75.3**	-6.2
Consis-Verb	74.0	72.2	70.7	69.7	-4.3
Consis-Verb-Topk	76.7	75.6	73.7	72.8	-3.9
(iii) Self-report on the returned answer					
Verb	68.2	65.2	62.1	59.9	-8.3
Verb-Topk	58.7	55.8	53.0	51.1	-7.6
(iv) Separate verification query					
P_{True}-Consis	72.3	71.4	70.6	69.8	-2.5
P_{True}-Prob	76.0	75.1	74.5	73.7	-2.3
P_{True}-Consis-Cand	68.9	67.4	66.6	66.1	-2.8
P_{True}-Prob-Cand	77.5	75.8	74.0	72.9	-4.6
Self-Ask	65.6	65.1	64.7	64.3	-1.3
Self-Ask-Cand	73.3	71.6	69.2	68.0	-5.3
Cardinality-rescaled (§6, verified count)					
CARE	78.1	78.1	77.0	76.4	-1.7

(c) Scores with no probability scale. The entropy-based scores are monotone transforms $\exp(-H)$ pushed into $[0, 1]$ by construction, which fixes their range without giving them a referent. A calibration number for this group therefore presumes a monotone map onto probabilities, and what such a map can repair is examined below.

The signed gap. Comparing a score's mean with accuracy avoids the binning of ECE and separates the two quantities that move. Table 27 reports mean confidence minus accuracy at each cardinality. Accuracy rises from 48.0 to 61.7 while every score falls, so the gap falls for all 15 baselines, by 14

Table 24: **AUROC on Qwen2.5-32B-Instruct under cumulative answer-cardinality mixtures.**

Method	$\leq$ **1**	$\leq$ **2**	$\leq$ **4**	$\leq$ **6**	**Δ**
(i) Whole predictive distribution					
Prob Entropy	77.8	74.4	72.8	71.3	−6.5
N-Prob Entropy	78.2	74.5	73.2	71.8	−6.4
Sem Entropy	73.7	70.4	70.0	69.2	−4.5
(ii) Support of the returned answer					
Consis	75.1	72.2	71.9	71.0	−4.1
Perplexity	**79.0**	75.7	74.5	72.9	−6.1
Consis-Verb	73.5	70.5	69.4	68.7	−4.8
Consis-Verb-Topk	75.6	73.6	72.3	71.0	−4.6
(iii) Self-report on the returned answer					
Verb	63.2	60.1	59.3	58.6	−4.6
Verb-Topk	59.3	55.7	51.6	49.5	−9.8
(iv) Separate verification query					
P_{True}-Consis	69.4	69.7	69.3	69.1	−0.3
P_{True}-Prob	78.3	**77.2**	**76.6**	**76.3**	−2.0
P_{True}-Consis-Cand	66.7	66.5	65.9	65.9	−0.8
P_{True}-Prob-Cand	78.1	75.4	74.0	73.7	−4.4
Self-Ask	71.8	69.9	67.6	66.3	−5.5
Self-Ask-Cand	71.8	69.1	67.1	65.9	−5.9
Cardinality-rescaled (§6, verified count)					
CARE	75.1	74.0	74.2	73.9	−1.2

Table 25: **AUROC on Qwen2.5-72B-Instruct under cumulative answer-cardinality mixtures.**

Method	$\leq$ **1**	$\leq$ **2**	$\leq$ **4**	$\leq$ **6**	**Δ**
(i) Whole predictive distribution					
Prob Entropy	**81.5**	78.3	77.4	**76.5**	−5.0
N-Prob Entropy	80.2	77.4	76.1	75.3	−4.9
Sem Entropy	76.4	74.7	74.2	73.0	−3.4
(ii) Support of the returned answer					
Consis	77.8	76.3	75.8	74.8	−3.0
Perplexity	80.8	**78.8**	**77.5**	76.4	−4.4
Consis-Verb	77.2	75.3	74.7	73.4	−3.8
Consis-Verb-Topk	74.6	72.4	71.3	70.1	−4.5
(iii) Self-report on the returned answer					
Verb	63.2	59.5	57.3	56.3	−6.9
Verb-Topk	67.1	63.3	59.0	57.1	−10.0
(iv) Separate verification query					
P_{True}-Consis	72.7	69.0	68.1	68.1	−4.6
P_{True}-Prob	79.8	75.6	74.5	74.4	−5.4
P_{True}-Consis-Cand	70.4	68.4	67.4	67.2	−3.2
P_{True}-Prob-Cand	80.2	76.0	74.4	74.1	−6.1
Self-Ask	66.2	63.1	61.1	60.6	−5.6
Self-Ask-Cand	72.8	68.9	65.6	64.2	−8.6
Cardinality-rescaled (§6, verified count)					
CARE	77.8	78.1	78.6	77.9	+0.1

to 39 points. *Consis* enters well calibrated (+3.3) and leaves severely underconfident (−25.8), and *Verb-Topk* crosses from +33.9 to −4.9. The self-report and post-hoc groups begin 16 to 47 points overconfident and remain overconfident throughout, their gap merely shrinking. The direction is the same for all of them, which is what §5.3 predicts: as valid mass spreads, every score that reads the generation falls while accuracy rises.

ECE and discrimination move in opposite directions. That shrinking is why a mixture-level ECE misleads here. An estimator that was overconfident on single-answer questions is met halfway by

Table 26: **AUROC on LLaMA-3.1-8B-Instruct under cumulative answer-cardinality mixtures.**

Method	≤ 1	≤ 2	≤ 4	≤ 6	Δ
(i) Whole predictive distribution					
Prob Entropy	69.3	67.5	63.0	60.5	-8.8
N-Prob Entropy	66.5	63.7	60.0	57.9	-8.6
Sem Entropy	77.7	77.3	74.2	72.6	-5.1
(ii) Support of the returned answer					
Consis	**80.3**	**80.0**	**77.1**	**76.5**	-3.8
Perplexity	**80.3**	78.9	75.8	74.8	-5.5
Consis-Verb	77.1	76.4	74.5	73.7	-3.4
Consis-Verb-Topk	77.9	76.7	74.7	73.7	-4.2
(iii) Self-report on the returned answer					
Verb	56.5	56.8	55.2	54.3	-2.2
Verb-Topk	56.7	56.2	54.7	53.7	-3.0
(iv) Separate verification query					
P_{True}-Consis	72.0	70.0	68.1	67.2	-4.8
P_{True}-Prob	74.5	71.5	69.1	68.2	-6.3
P_{True}-Consis-Cand	72.0	70.0	68.0	67.4	-4.6
P_{True}-Prob-Cand	72.6	70.6	68.8	68.0	-4.6
Self-Ask	66.8	63.9	62.8	62.9	-3.9
Self-Ask-Cand	76.0	74.5	72.7	72.1	-3.9
Cardinality-rescaled (§6, verified count)					
CARE	80.3	81.1	79.9	79.7	-0.6

rising accuracy, so its calibration error falls for a reason unrelated to the quality of its ordering: from cardinality 1 to 6, ECE improves for 10 of the 15 baselines while all 15 lose AUROC. *Verb-Topk* is the clearest case, improving from 35.9 to 21.9 ECE while its AUROC falls from 60.0 to 48.9, which is chance. *Perplexity* improves from 30.6 to 19.6 while losing 12.8 AUROC points. Table 28 reports ECE over the cumulative mixtures for reference. Across estimators the two metrics agree only moderately (Spearman $\rho = -0.64$ between pooled ECE and pooled AUROC) and they invert on individual pairs: *Verb-Topk* records a pooled ECE 10.5 points better than P_{True}-*Prob-Cand* (23.4 against 33.9) and an AUROC 22.2 points worse (48.4 against 70.6). A calibration number is therefore not a substitute for a discrimination number in this setting, and can point the opposite way.

What recalibration can and cannot repair. Two facts bound what recalibration can do. Platt scaling is strictly monotone, so fitting it leaves every AUROC in Table 3 exactly unchanged: the discrimination loss of §5.2 is not a calibration problem. A single strictly increasing transformation applied uniformly to every question cannot change that loss, although cardinality-dependent corrections such as CARE can. The map is also cardinality-dependent. Fitting Platt at cardinality 1 and applying it at cardinality 6 costs a median 6.6 ECE points against a map fitted at cardinality 6, for 14 of the 15 baselines, reaching 16.4 for *Verb-Topk* and roughly 12 points for both *Consis* and *Perplexity*; cardinality is unknown at query time, so the better map is unavailable. *Self-Ask* is the exception, its scores concentrating on a few discrete values that leave a two-parameter fit unstable. Aligning scores by rank within each cardinality—again using cardinality that a deployed system does not have—recovers 30 to 45 percent of the pooled AUROC loss (45 for *Consis*, 45 for *Sem Entropy*, 30 for *Perplexity*), placing roughly half of that loss in the cross-cardinality level shift and the rest inside individual cardinalities. The shift is the part a count addresses (§6); the remainder is not reachable by any monotone map.

D.8 Why Does Confidence Decline? A Per-Method Analysis

§5.3 attributes the decline to an explicit or implicit dependence on how support is divided among valid answers. This section gives the per-estimator evidence for that account, taking the groups of §4 in turn: sample-agreement scores and entropies first, then generation probabilities, then the verbalized and verification scores that read no probabilities at all.

Table 27: **Signed calibration gap** (mean confidence minus accuracy, in points) at each answer cardinality for LLaMA-3.1-70B-Instruct. Accuracy rises from 48.0 at cardinality 1 to 55.4, 59.9 and 61.7, so a falling gap means a score that grows underconfident as valid answers are added. The gap falls for all 15 estimators.

Method	1	2	4	6	Δ
(i) Whole predictive distribution					
Prob Entropy	+23.6	+14.6	+6.7	+3.8	−19.8
N-Prob Entropy	+43.5	+35.5	+30.0	+28.0	−15.5
Sem Entropy	+12.8	+2.5	−10.2	−16.6	−29.3
(ii) Support of the returned answer					
Consis	+3.3	−6.6	−19.7	−25.8	−29.1
Perplexity	+29.6	+21.0	+13.1	+9.2	−20.4
Consis-Verb	+2.3	−7.4	−19.4	−25.3	−27.6
Consis-Verb-Topk	−0.4	−9.5	−21.5	−27.5	−27.1
(iii) Self-report on the returned answer					
Verb	+46.7	+39.2	+33.2	+30.5	−16.2
Verb-Topk	+33.9	+18.3	+2.1	−4.9	−38.7
(iv) Separate verification query					
P_{True}-Consis	+31.3	+23.8	+17.1	+13.8	−17.5
P_{True}-Prob	+31.2	+23.6	+16.9	+13.7	−17.5
P_{True}-Consis-Cand	+40.1	+32.5	+27.2	+26.1	−13.9
P_{True}-Prob-Cand	+40.1	+32.4	+27.2	+26.1	−14.0
Self-Ask	+15.6	+5.4	−3.8	−7.3	−22.9
Self-Ask-Cand	+16.5	+8.0	−0.4	−5.0	−21.5

Table 28: **ECE under cumulative answer-cardinality mixtures** for LLaMA-3.1-70B-Instruct, in points, lower better. ΔAUROC repeats Table 3. ECE improves over these mixtures for 11 of the 15 estimators while all 15 lose AUROC, so the two metrics disagree on the direction of the effect.

Method	[1]	[1,2]	[1,2,4]	[1,2,4,6]	ΔECE	ΔAUROC
(i) Whole predictive distribution						
Prob Entropy	26.2	22.4	20.5	20.0	−6.2	−8.5
N-Prob Entropy	43.5	39.5	36.4	34.3	−9.3	−10.1
Sem Entropy	13.2	9.7	9.5	11.3	−1.8	−6.8
(ii) Support of the returned answer						
Consis	8.7	7.9	11.4	14.9	+6.2	−5.6
Perplexity	30.6	25.9	23.0	21.5	−9.1	−7.1
Consis-Verb	11.7	10.4	12.8	15.7	+4.1	−5.8
Consis-Verb-Topk	12.4	12.5	15.1	18.2	+5.9	−5.8
(iii) Self-report on the returned answer						
Verb	46.8	43.1	40.3	38.5	−8.4	−2.5
Verb-Topk	35.9	29.1	25.0	23.4	−12.5	−11.6
(iv) Separate verification query						
P_{True}-Consis	35.8	34.0	32.9	32.2	−3.6	−5.2
P_{True}-Prob	35.1	33.8	32.5	31.8	−3.3	−6.3
P_{True}-Consis-Cand	41.7	38.4	35.4	34.1	−7.6	−0.1
P_{True}-Prob-Cand	41.6	38.3	35.4	33.9	−7.8	−4.5
Self-Ask	36.7	38.7	41.0	42.4	+5.7	−2.7
Self-Ask-Cand	26.3	24.4	23.8	24.0	−2.4	−0.3

Sample-agreement scores and entropies (groups (i) and (ii)). *Consis* and *Sem Entropy* estimate confidence by measuring agreement among multiple sampled responses. Case-level inspection shows that the decline is directly attributable to **valid answer diversity fragmenting the generation probability mass**. When a question has a single valid answer, sampled responses 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, asked *"Name one person who received the Nobel Prize in Physics in 1995,"* the model alternates between "Martin Lewis Perl" and "Frederick Reines"—both correct—across samples. Consistency-

Table 29: **Verbalized confidence divides self-reported mass across valid answers** (Verb-Topk, LLaMA-3.1-70B-Instruct; averages over all 12,000 questions). The model is prompted for five candidate answers with a confidence for each. *Valid listed* counts how many of the listed candidates are in the answer set; *stated conf. on valid* sums the confidences assigned to them; *returned answer* is the confidence stated for the answer the model returned, and *its share* that confidence divided by *stated conf. on valid*.

	1a	**2a**	**4a**	**6a**
Candidates listed	3.92	4.38	4.79	4.94
Valid listed	0.70	1.17	1.95	2.45
Stated conf. on valid	0.45	0.52	0.57	0.58
Returned answer	0.81	0.74	0.62	0.57
its share	0.75	0.69	0.58	0.52

based methods read this diversity as disagreement, producing a low score that is indistinguishable from the low score caused by genuine uncertainty, i.e. alternation between correct and incorrect answers. This indistinguishability, not the low score itself, is the failure mode.

Generation probabilities (group (ii)). For generation-probability methods such as *Perplexity*, the decline may appear surprising, since these methods operate on a single greedy or sampled response rather than on multi-sample agreement. To locate the cause, we measure the entropy of the **first generated token's top-20 logits** on LLaMA-3.1-70B-Instruct, stratified by answer count (Table 30). The monotonic increase confirms that even at the onset of generation, the model's probability mass is more dispersed when more valid answers exist. The model "hesitates" among multiple plausible starting tokens (e.g., "Martin" vs. "Frederick"), each leading to a different correct answer. This initial dispersion propagates through the sequence, lowering overall sequence probability and thus Perplexity-based confidence. The same answer-diversity mechanism that affects consistency-based methods therefore also affects probability-based methods, but manifests at the *token distribution* level rather than the *sample agreement* level.

Self-report (group (iii)). Verbalized confidence is elicited in words and never reads the token distribution, yet it declines for the same structural reason, which is the implicit dependence of §5.3. Verb-Topk prompts the model for its top five answers with a confidence for each, which makes the mechanism directly observable (Table 29): as cardinality grows the model lists more *valid* answers, and the confidence it distributes over them rises, closely tracking the valid probability mass q_π of Table 2. What falls is the share retained by the answer it returned. The reported score is therefore a self-reported analogue of $q\,r_j$, produced by the model's own judgment rather than by its logits. Plain Verb, which asks for a single answer and a single confidence, has no alternatives to divide mass among; it stays close to saturation across cardinalities and consequently carries little discriminative signal at any of them (58.0 AUROC at [1], Table 3). The two Consis-Verb variants use confidence-weighted agreement, so they inherit the dispersion sensitivity of both channels and show substantial confidence declines (-13.9 and -13.4 points, Table 1).

Separate verification queries (group (iv)). Verification queries show the smallest decline of the four groups. In Table 1, P_{True}-Consis-Cand reports 88.1, 87.9, 87.1 and 87.9 from 1a through 6a, a change of 0.2 points. The robustness follows from **decoupling confidence estimation from the generation process**: rather than reading confidence off 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 rely on the model's *judgment*. Verifying a single correct answer does not require choosing among alternatives, so the verification step is largely insensitive to how many other valid answers exist. This advantage costs an additional inference pass per question, and greater stability does not automatically translate into higher AUROC (Table 3): verification queries remain the flattest group without overtaking the returned-answer scores in absolute discrimination, because the verification query trades a cardinality-dependent signal for a weaker one. Nor does supplying the candidates guarantee stability: P_{True}-Prob-Cand reads the verdict token's probability and still loses 4.5 AUROC points, the asymmetry §5.3 notes.

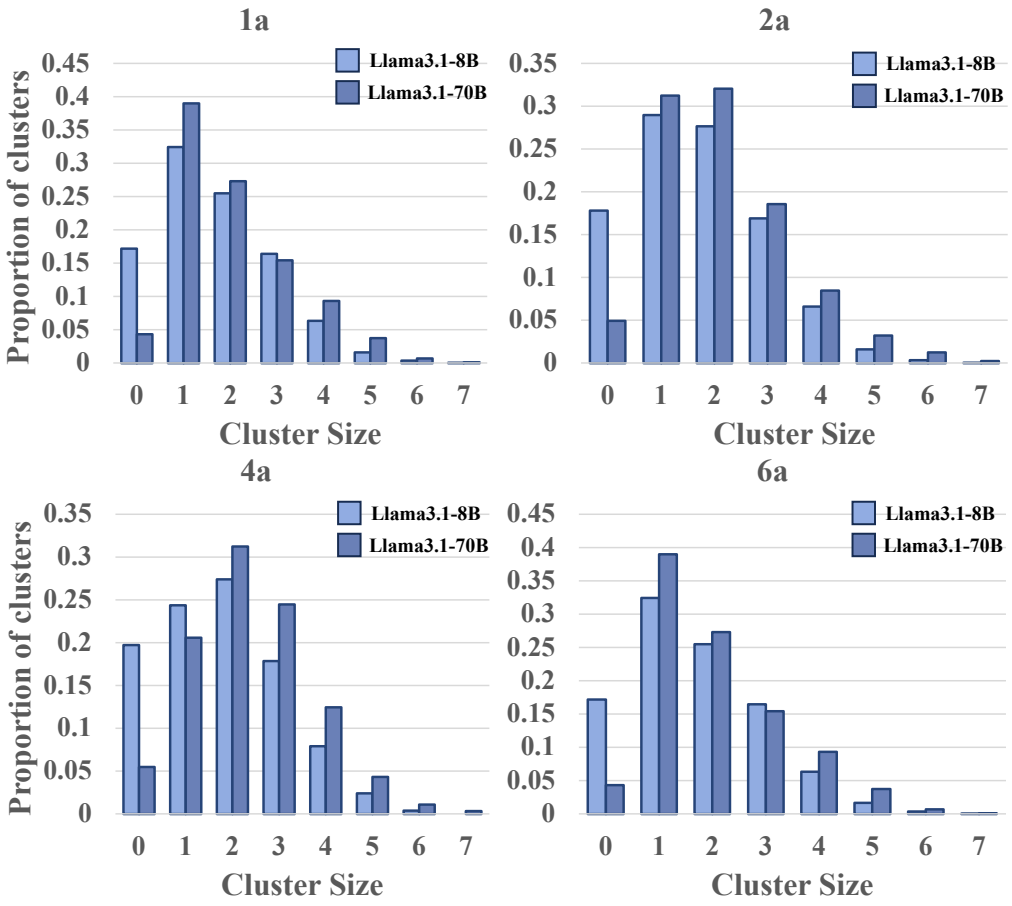

Figure 7: **Distribution of cluster sizes for the LLaMA family.**

Table 30: **First-token entropy of LLaMA-3.1-70B-Instruct** under different answer cardinalities.

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

D.9 Entropy Decomposition

The entropy decomposition in this section follows from grouping the model's semantic modes into the valid set V and the invalid set I. Writing $q = \sum_{m \in V} \pi_m$ and $\pi_V(m) = \pi_m / q$ for $m \in V$ (analogously for I), the chain rule for entropy over the two-stage choice of first a group and then a mode within it gives

$$H(\pi) = h(q) + q\, H(\pi_V) + (1 - q)\, H(\pi_I),$$

where h is the binary entropy. The first term is the uncertainty about whether the model will produce a valid answer at all—the only component aligned with the prediction target—while the remaining two measure how finely the model splits its mass *within* the valid and invalid groups.

Two implementation notes. The mode distribution π is estimated from 20 samples, so $H(\pi)$ is a finite-sample estimate and is not identical to the token-level predictive entropy an estimator computes; we use it because the decomposition requires assigning validity labels to modes, which token-level entropy does not expose. And a mode is labeled valid by majority vote among its members, so a cluster mixing valid and invalid surface forms is assigned entirely to one side. Both approximations

affect the absolute magnitudes rather than the comparison across cardinalities, which is what the decomposition is used for. Table 31 gives the three terms on MACE.

Table 31: **Entropy decomposition** for LLaMA-3.1-70B-Instruct, in nats. Reported entropy grows by 0.284, of which 0.313 is valid-answer diversity, while error diversity falls.

	1a	**2a**	**4a**	**6a**	Δ
$H(\pi)$ total	0.687	0.732	0.880	0.971	$+0.284$
$h(q)$ correctness	0.152	0.176	0.223	0.229	$+0.077$
$q\,H(\pi_V)$ valid div.	0.020	0.095	0.240	0.333	$+0.313$
$(1-q)H(\pi_I)$ error div.	0.516	0.462	0.418	0.409	-0.106
$H^+ = H(\pi_V)$	0.04	0.15	0.35	0.47	$+0.43$
$r_{\max}$	0.977	0.921	0.840	0.789	-0.188

D.10 Correlation Between Confidence and Knowledge Coverage

Validity of Spearman Correlation. Spearman's rank correlation is sensitive to tied ranks and can be misleading when one variable is heavily concentrated at a single value. We therefore verify that knowledge coverage (the number of semantic clusters retained after filtering) is sufficiently dispersed to support a rank-based analysis.

Figure 7 shows the distribution of cluster counts across models and all four ground-truth settings (`1a/2a/4a/6a`) after applying $\tau = 0.1$. Two observations support the validity of the analysis:

1. **No single cluster count dominates.** In every setting, counts span 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 count, which would produce excessive tied ranks and distort the coefficient.
2. **Systematic shift with increasing ground-truth cardinality.** As the number of ground-truth answers grows from $k = 1$ to $k = 6$, the distribution shifts smoothly toward larger values without collapsing into a narrow mode or generating extreme outliers. This gradual expansion indicates that the effective answer space grows stably, providing a well-graded ordinal variable.

Correlation Results. Table 33 reports the Spearman correlation between model confidence and knowledge coverage using the *consistency* method. Across most models we observe a consistent **negative correlation**: as coverage expands (more semantically distinct modes survive filtering), reported confidence *decreases*. Coverage itself grows with cardinality for every model (Figure 8). Higher confidence therefore does not indicate broader factual coverage; models are most confident when their output distribution concentrates on a small number of semantic modes, even when a wider set of valid answers exists.

Notably, **larger models exhibit stronger negative correlation** than smaller counterparts, indicating that as capacity increases, the confidence score becomes *more* sensitive to expansion of the effective answer space—precisely the regime where the loss of correctness discrimination is most consequential. The size of the decline follows the same within-family ordering, which is the observation §5.3 reports. Taking *Consis* from the per-model tables of §D.1, the drop from 1a to 6a grows with parameter count at every step in Qwen2.5 (1.2, 1.7, 3.7 and 5.6 points from 7B to 72B) and from 5.6 to 15.4 points between LLaMA-3.1-8B and 70B; among the API models the larger GPT-4o loses 17.5 points against 3.6 for GPT-4o-mini. Scale therefore does not mitigate the mismatch between concentration and correctness under multiple valid answers but sharpens it, consistent with a broader set of valid answers leaving more room for support to disperse.

Valid-mode dispersion and the size of the confidence drop. The account in §5.3 predicts something more specific than a correlation with coverage: the models penalized most should be those whose *valid*-mode dispersion grows most as cardinality increases. Table 32 reports both quantities for the six open-weight models. Growth in valid-mode entropy from 1a to 6a ranges from 0.088 (Qwen2.5-14B) to 0.434 (LLaMA-3.1-70B), and it tracks the drop in consistency-based confidence closely across models (Pearson $r = 0.93$). Two caveats bound how much weight this can carry. Six model points cannot establish a scaling relationship, and the association is driven largely by the

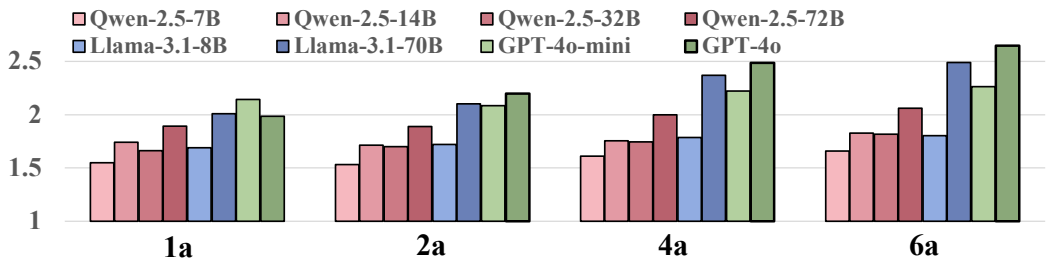

Figure 8: **Average knowledge coverage across ground-truth (GT) set sizes.** Different color schemes denote model families, with darker shades indicating larger model scales.

Table 32: **Models whose valid-mode dispersion grows more show larger confidence drops.** M^+ is the number of valid semantic answer modes covered and H^+ the entropy over them (nats); ΔConf is the change in *Consis* from 1a to 6a in points, as reported per model in §D.1.

Model	M^+ **(1a)**	M^+ **(6a)**	H^+ **(1a)**	ΔH^+	Δ **Conf**
Qwen2.5-7B-Instruct	0.49	1.24	0.015	$+0.108$	-1.2
Qwen2.5-14B-Instruct	0.51	1.16	0.009	$+0.088$	-1.7
Qwen2.5-32B-Instruct	0.52	1.25	0.009	$+0.102$	-3.7
Qwen2.5-72B-Instruct	0.60	1.42	0.012	$+0.134$	-5.6
LLaMA-3.1-8B-Instruct	0.72	2.37	0.026	$+0.268$	-5.6
LLaMA-3.1-70B-Instruct	0.89	3.01	0.036	$+0.434$	-15.4

difference between the two families: within Qwen2.5, H^+ growth is not monotone in parameter count (0.108, 0.088, 0.102, 0.134 for 7B through 72B), although the confidence drop is. We therefore treat this as evidence consistent with the proposed mechanism rather than as a scaling law, and note that families differ in how broadly they sample at a fixed temperature, which is itself a plausible contributor.

D.11 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 Unnormalized Cluster Probabilities. A key design decision in our semantic-clustering pipeline is to measure each cluster's importance by its **unnormalized cumulative probability**, defined as the sum of sequence-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 renormalizing these probabilities across clusters (i.e., converting them into a simplex distribution $\hat{P}(C_m) = P_{\text{raw}}(C_m) / \sum_m P_{\text{raw}}(C_m)$). Normalization 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), normalization would still assign some cluster a large *relative* share, creating the false impression of a confident semantic mode. Operating on unnormalized probabilities preserves the absolute scale of the model's output distribution, so that a cluster's importance directly reflects the model's genuine probability commitment.

Empirical Distribution and Threshold Selection. Figure 5 shows the distribution of $P_{\text{raw}}(C_m)$ across all clusters produced by **LLaMA-3.1-70B-Instruct**. The distribution is clearly **bimodal**:

Table 33: **Spearman correlation of confidence with knowledge coverage** across ground-truth (GT) set sizes using the *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

- A dominant spike in the $[0,\ 0.1)$ bin, accounting for over **70%** of all clusters. These low-mass clusters each capture only a tiny fraction of the 20 samples—typically one or two responses with individually low generation probabilities. Manual inspection confirms 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 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 is threefold:

1. **Empirical gap alignment.** The threshold sits at the boundary between the dominant near-zero spike and the sparsely populated mid-range bins. Any value in $[0.05,\ 0.15]$ produces similar filtering outcomes; 0.1 is 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. We use this threshold only as an empirical filter for low raw-mass fragmented clusters; it is not interpreted as a fixed fraction of the total probability budget.
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.

Applying this threshold yields a compact set of *effective* semantic clusters that represent the model's meaningful output modes, filtering out sampling noise and providing a cleaner basis for measuring knowledge coverage.

D.12 The Answer Count: Correction Form, Required Accuracy, and Where It Could Come From

Why the correction saturates. Two ways of applying a count to a frequency are natural, and they behave very differently (Table 34). Treating the count as a number of independent chances, $1 - (1 - c)^K$, gains 4.3 AUROC points in the broadest mixture. Treating it as a multiplier, Kc, gains 1.9 points. The direct multiplier and the saturating form therefore behave differently in this experiment; the comparison does not establish that unboundedness alone causes the lower AUROC. The saturating form is also far more tolerant of an imperfect count, which matters more than the ceiling. With the best count estimate we can build it still gains 1.1 points, whereas the multiplicative form turns negative (-1.0): an overshoot that is harmless when K is right becomes damaging when K is wrong.

How accurate the count has to be. Equation 1 uses the numerical value of K in the rescaling, so an ordering-preserving perturbation need not preserve the final scores. We therefore assess sensitivity to count errors under a controlled perturbation scheme. The observed break-even correlation characterizes this scheme rather than a universal requirement for cardinality estimation. We measure the

Table 34: **How the count is applied**, on LLaMA-3.1-70B-Instruct over 100% of MACE. Both forms rescale the same quantity, the fraction c of the 20 samples agreeing with the returned answer, whose uncorrected AUROC is the first row. Δ is the change in the broadest mixture against that row.

Score	$\leq$ **1**	$\leq$ **2**	$\leq$ **4**	$\leq$ **6**	**Δ**
Consis, uncorrected	80.1	79.7	77.3	74.6	—
CARE, $1-(1-c)^K$, true count	80.1	80.8	80.6	**78.9**	+4.3
Multiplicative, Kc, true count	80.1	79.6	78.6	76.4	+1.9
CARE, best label-free count estimate	78.9	78.9	77.6	75.6	+1.1
Multiplicative, same estimate	73.5	74.9	74.9	73.6	−1.0

requirement directly by blending the true count with a random permutation of itself and sweeping the blend weight, which traces gain against rank correlation (Table 35). The correction breaks even at Spearman $\rho \approx 0.36$, returns 1.3 of its 4.3 points at $\rho = 0.55$, 2.6 at $\rho = 0.82$, and 3.8 at $\rho = 0.96$. Below break-even it is actively harmful, since rescaling by a count that is uncorrelated with the truth injects noise into an otherwise usable score (-3.0 at $\rho = 0$). The requirement is therefore a threshold rather than a slope, and it is the number any future count estimator should be reported against.

Table 35: **What CARE returns as a function of how well the count is ranked**, broadest mixture. The count is blended with a random permutation of itself at decreasing weight; each row is the mean of five draws.

Spearman ρ with the true count	0.00	0.30	0.55	0.82	0.96	0.97	1.00
AUROC, $[1, 2, 4, 6]$	71.5	73.9	75.9	77.2	78.4	78.8	**78.9**
Δ over Consis	−3.0	−0.6	+1.3	+2.6	+3.8	+4.2	+4.3

Where the count could come from. We report what we could build without correctness labels and why we do not propose it as a method. Fitting a regressor on thirteen signals—distribution shape, mode counts, margins, the verbalized self-report and the post-hoc query—with domains held out reaches $\rho = 0.392$ (MAE 1.50), just past break-even, and converts to the $+1.1$ of Table 34. The reason these features are weak is structural rather than a matter of model capacity: every one of them reads dispersion, and dispersion is exactly the quantity that the valid answers and the model's own doubt both produce, so a question with four valid answers the model knows and a question with one valid answer the model is unsure of present the same evidence. The count, however, is not a property of the distribution at all. It is fixed by the question before the model is queried—by how wide a constraint the question imposes, how large the class it refers to is, and what type of entity it asks for—and none of that depends on the decoder, the sampling temperature, or whether the model knows any of the answers. Estimating it from the question rather than from the output distribution is the direction we would take, and Table 35 states the accuracy such an estimator must reach to be worth deploying.