LLM UNCERTAINTY REVEALED IN BEHAVIORAL BOUNDARY EFFECTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models may produce fluent and confident answers even when those answers are unreliable. Existing black-box uncertainty estimators typically rely on repeated sampling from the same model or disagreement across an unordered collection of models. The former may fail when the model consistently reproduces the same error, while the latter may introduce unnecessary disagreement and inference cost due to differences in model architecture and training distribution. In this work, we investigate whether an ordered family of small models can serve as a more effective uncertainty probe, and propose **AnchorAUC**. Given an input, a target model first generates an anchor answer, after which a sequence of progressively smaller probe models independently answers the same input. Their semantic retention relative to the anchor forms a directional degradation trajectory along an ordered model-scale axis. AnchorAUC integrates this trajectory into a reliability score: answers that remain stable across a broader range of probe levels receive lower uncertainty, whereas answers that drift earlier exhibit a behavioral boundary associated with higher uncertainty. We evaluate AnchorAUC on five question-answering benchmarks, four target models, and two probe families. AnchorAUC achieves competitive AUROC and AUARC across diverse model–task configurations while requiring substantially lower inference cost than multi-anchor sampling. Further analyses of probe order, correctness transitions, and probe granularity show that the ordered trajectory contains information beyond generic cross-model disagreement.

1 INTRODUCTION

Large language models may generate fluent and confident responses even when their answers are unreliable (Lin et al., 2022; Kalai & Vempala, 2024; Huang et al., 2025). A common black-box approach to identifying such responses is to measure *within-model consistency*: the same model is sampled repeatedly, and uncertainty is estimated from variations among its outputs (Manakul et al., 2023; Kuhn et al., 2023; Farquhar et al., 2024; Arana et al., 2026). However, when a model consistently reproduces the same incorrect answer, repeated sampling may still yield highly similar responses and consequently underestimate uncertainty.

Cross-model comparison provides additional external evidence by examining whether other models generate compatible answers (Hamidieh et al., 2026). Existing methods, however, typically treat the participating models as an unordered collection. Because different models may vary substantially in architecture and training distribution, their disagreement can reflect family-specific behavioral differences rather than the reliability of the answer itself. Repeatedly querying multiple models also incurs considerable inference cost. These limitations motivate a more controlled way to organize cross-model answer behavior.

We observe that a scale-ordered family of smaller models often exhibits structured changes in its responses to the same input. As illustrated in Figure 1, given a question and an answer produced by a larger model as the anchor, we compare the outputs of smaller probe models with the anchor to measure the degree of semantic drift. Answers with broader behavioral support tend to remain stable across multiple model levels, whereas less stable answers are more likely to drift as the probe models become smaller. The outputs across these model levels therefore form a directional degradation trajectory. This trajectory allows us to more clearly observe a potential *behavioral boundary*: the point along the decreasing probe sequence at which a previously stable answer begins to exhibit

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.

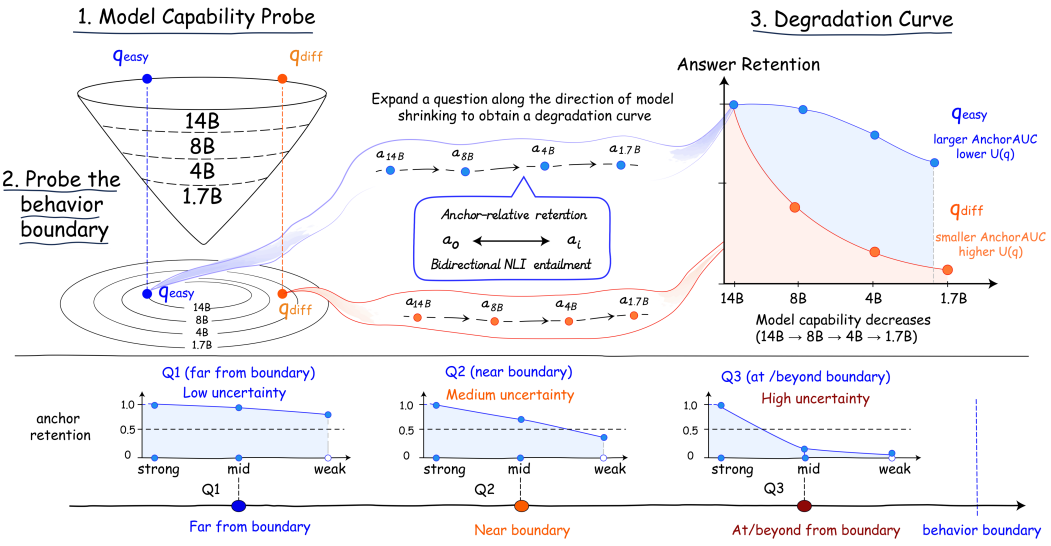


Figure 1: Ordered small-model probing for black-box uncertainty estimation.

substantial semantic drift. Such changes may arise from factual knowledge, reasoning, context utilization, or other factors, but regardless of their source, they constitute an observable manifestation of uncertainty.

Based on this idea, we propose **AnchorAUC**, a black-box uncertainty estimation method based on an ordered family of small models. The target model first generates an anchor answer, after which a sequence of progressively smaller probe models independently answers the same input. Each probe answer is then compared with the anchor answer. The resulting retention scores jointly form an answer-retention degradation curve, whose area is used as the reliability score. Answers that remain stable across a wider range of probes are assigned lower uncertainty, whereas answers that cross the behavioral boundary earlier are assigned higher uncertainty.

Ordered small-model probes preserve the directional structure discarded by unordered cross-model disagreement, while each lightweight probe requires only one generation. Compared with repeated sampling from multiple anchor-scale models, they can therefore provide a structured uncertainty signal at substantially lower inference cost (Dubey et al., 2024; Yang et al., 2025; Cui et al., 2026). The main contributions of this work are:

- **Ordered small-model families as uncertainty probes.** We organize cross-model consistency along a decreasing model-scale sequence, transforming unordered model disagreement into a structured answer-retention trajectory.
- **AnchorAUC, a black-box uncertainty estimator.** AnchorAUC measures how well an anchor answer is retained across the ordered probe sequence and summarizes the resulting degradation curve through its area.
- **Systematic validation of the ordered probe signal.** Across diverse QA benchmarks, AnchorAUC achieves competitive AUROC and AUARC while substantially reducing inference cost relative to multi-anchor sampling. Probe-order assignments, count-preserving nulls, probe-granularity analyses, and large-anchor experiments show that the ordered trajectory contains information beyond generic cross-model disagreement.

2 Related Work

Black-box uncertainty estimation evaluates the reliability of generated answers using only observable model behavior, without requiring access to token probabilities, hidden representations, or other model-internal information. A common approach is to measure *within-model consistency*: the same model is sampled multiple times, and disagreement among its responses is used as an uncertainty

signal. SelfCheckGPT compares a target response against stochastic samples and relates factual inconsistency to hallucination risk (Manakul et al., 2023). Semantic uncertainty further accounts for linguistic variability by grouping semantically equivalent generations before estimating uncertainty (Kuhn et al., 2023; Farquhar et al., 2024). Subsequent work has extended this paradigm through semantic-space density, pairwise similarity structures, relevance-aware scoring, and alternative entropy estimators (Qiu & Miikkulainen, 2024; Nikitin et al., 2024; Aichberger et al., 2025). Other methods derive confidence scores from observable generations or use external models to assess answer consistency (Lin et al., 2024; Chen & Mueller, 2024). More recently, variability in final answers across repeated stochastic generations has itself been used as an uncertainty signal (Arana et al., 2026). Although these methods effectively characterize generation variability, they may still underestimate uncertainty when a model consistently reproduces the same incorrect answer.

Cross-model consistency complements within-model sampling by comparing the target answer with outputs from other models (Hamidieh et al., 2026). However, existing methods usually treat these models as an unordered collection, so disagreement caused by differences in architecture or training distribution may be mixed with disagreement that is informative about answer reliability. Querying several models of comparable scale can also be costly. AnchorAUC instead uses a within-family sequence of progressively smaller models and measures how well a fixed anchor answer is retained along this order. The resulting directional trajectory captures not only whether disagreement occurs, but also where semantic drift begins, providing an observable characterization of a *behavioral boundary*.

3 METHOD

3.1 ORDERED SMALL-MODEL PROBE FAMILY

Let the input be $x = (q, c)$, where q is a question and c is an optional context. A target model M_0 first generates the anchor answer $a_0 = M_0(x)$. We then construct an ordered family of smaller probe models

$$\mathcal{P} = \{M_1, M_2, \ldots, M_K\}, \qquad s_1 > s_2 > \cdots > s_K, \tag{1}$$

where s_i denotes the parameter scale of probe M_i. Each probe independently generates $a_i = M_i(x), i = 1, \ldots, K$. We order the selected probes by model scale to form a reproducible sequence from larger to smaller models. The probe models are selected from a common model family to reduce additional variation caused by differences in architecture, training distribution, and instruction tuning. The target model may come from another family, while the probe sequence itself remains an ordered within-family scale hierarchy.

3.2 ANCHOR-RELATIVE ANSWER RETENTION

For each probe answer a_i, we measure its semantic consistency with the anchor answer a_0. Because one-directional entailment may remain high when one answer contains only part of the information expressed by the other, we use bidirectional natural language inference. Let $E(u \Rightarrow v) = P_{\text{NLI}}(\text{entailment} \mid u, v)$ denote the probability that u entails v. The anchor-relative retention score is

$$r_i = \frac{1}{2}\left[E(a_0 \Rightarrow a_i) + E(a_i \Rightarrow a_0)\right], \qquad i = 1, \ldots, K, \tag{2}$$

with $r_0 = 1$. Combining the two directions penalizes both the omission of anchor content and the introduction of content not supported by the anchor, yielding a symmetric retention score in $[0, 1]$.

For structured short answers, including binary answers, multiple-choice labels, and single numerical answers, we use strict normalized matching: identical answers receive a score of 1, and different answers receive 0. Bidirectional NLI is used for free-text answers.

3.3 ANSWER-RETENTION DEGRADATION CURVE

The retention profile $\mathcal{R}(x) = (r_0, r_1, \ldots, r_K)$ describes how well the anchor answer is retained as probe model scale decreases. It provides an observable behavioral characterization of scale-ordered

answer stability. In particular, the region along the ordered probe sequence where retention begins to decline substantially can be interpreted as an observable *behavioral boundary*.

To represent the nonuniform scale gaps within the selected probe family, let $\bar{s}_0$ denote the reference scale assigned to the anchor position. For target models with known parameter counts, we set $\bar{s}_0$ to the target-model scale. We then define

$$z_0 = 0, \qquad z_i = \frac{\bar{s}_0 - s_i}{\bar{s}_0 - s_K}, \qquad i = 1, \ldots, K. \tag{3}$$

Thus, the anchor is located at $z_0 = 0$, the weakest probe at $z_K = 1$, and $z_{i+1} - z_i$ preserves the relative raw-scale gap between adjacent probe levels.

We use parameter count as a heuristic coordinate to represent the relative spacing among model levels in the degradation sequence. When the target-model scale is known, its actual parameter count is used to locate the anchor. As shown in Appendix C, the raw-size coordinate consistently outperforms the logarithmic alternative and is therefore used as the default parameterization. For closed-source anchors with unavailable parameter counts, we instantiate the coordinate using explicitly specified fixed reference placements.

Given the points $\{(z_i, r_i)\}_{i=0}^{K}$, we define the degradation curve by linearly interpolating adjacent levels:

$$\widehat{r}_x(z) = \frac{z_{i+1} - z}{z_{i+1} - z_i} r_i + \frac{z - z_i}{z_{i+1} - z_i} r_{i+1}, \qquad z \in [z_i, z_{i+1}], \quad i = 0, \ldots, K-1. \tag{4}$$

3.4 AnchorAUC

AnchorAUC is defined as the area under the answer-retention degradation curve:

$$\begin{aligned} \mathrm{AnchorAUC}(x) &= \int_0^1 \widehat{r}_x(z)\, dz \\ &= \sum_{i=0}^{K-1} (z_{i+1} - z_i)\, \frac{r_i + r_{i+1}}{2}. \end{aligned} \tag{5}$$

Since the model-scale axis is normalized to $[0, 1]$ and $r_i \in [0, 1]$, AnchorAUC also lies in $[0, 1]$. A large AnchorAUC indicates that the anchor answer remains stable across a broad range of progressively smaller probes. A small value indicates that substantial semantic drift emerges earlier along the ordered sequence, corresponding to an earlier behavioral transition and higher uncertainty.

We treat AnchorAUC as an answer-reliability score and define the corresponding uncertainty as

$$U(x) = 1 - \mathrm{AnchorAUC}(x). \tag{6}$$

4 Experimental Setup

4.1 Datasets, Baselines, and Evaluation Metrics

We evaluate on five question-answering benchmarks. CoQA (Reddy et al., 2019) and HotpotQA (Yang et al., 2018) represent context-grounded QA; TriviaQA (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019) evaluate open-domain factual knowledge; and StrategyQA (Geva et al., 2021) contains binary questions requiring implicit multi-step reasoning. For each benchmark, we use a fixed set of $1{,}000$ examples from the evaluation split, with all methods evaluated on the same inputs and anchor answers.

We compare AnchorAUC with six representative baselines: P(True) (Kadavath et al., 2022), SAR (Duan et al., 2024), Semantic Entropy (SE) (Kuhn et al., 2023; Farquhar et al., 2024), $\mathrm{SE}_{\mathrm{SDLG}}$ (Aichberger et al., 2025), Self-Consistency (Self-Cons.), and Total Uncertainty (TU) (Hamidieh et al., 2026). These methods respectively capture self-evaluated confidence, relevance-aware uncertainty, semantic uncertainty, semantically diverse generation, repeated-generation agreement, and combined within- and cross-model disagreement. We report AUROC (Fawcett, 2006) for detecting incorrect anchor answers and AUARC (Nadeem et al., 2009) for selective prediction. Higher values indicate better uncertainty estimation.

Table 1: Performance comparison across multiple anchor models and probe configurations. Each cell reports AUROC/AUARC. Best results are bolded and second-best results are underlined.

Anchor: Qwen3-14B

Method	TriviaQA	StrategyQA	NQ	HotpotQA	CoQA	Average
	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	
P(True)	53.6/57.3	66.8/94.2	59.7/40.5	44.2/54.2	50.6/61.5	55.0/61.5
SAR	72.8/76.8	72.9/95.7	74.1/50.9	60.8/72.4	52.8/65.1	66.7/72.2
SE	65.7/70.3	73.4/95.9	71.2/48.3	55.7/68.3	51.2/62.2	63.4/69.0
SE_{SDLG}	74.7/70.9	79.4/98.4	78.1/49.5	64.8/72.5	50.3/66.2	69.5/71.5
Self-Cons.	75.0/86.7	74.6/96.1	50.5/42.1	56.8/88.4	60.2/85.9	63.4/79.8
TU	82.6/88.5	76.3/95.6	74.3/49.9	76.5/94.1	73.2/89.3	76.6/83.5
Probe-Qwen	84.6/89.6	**88.4/98.6**	73.3/49.8	**80.6/96.7**	74.7/90.9	80.3/85.1
Probe-Llama	**93.1/92.5**	79.8/97.2	**79.6/52.7**	76.3/94.9	**81.6/91.5**	**82.1/85.8**

Anchor: Phi-4-14B

Method	TriviaQA	StrategyQA	NQ	HotpotQA	CoQA	Average
	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	
P(True)	52.7/56.8	68.1/93.8	59.1/44.1	42.0/52.7	46.0/60.0	53.6/61.5
SAR	72.6/77.1	75.4/96.0	73.6/55.4	58.6/71.0	48.0/63.0	65.6/72.5
SE	64.6/69.7	75.8/96.2	70.5/52.6	53.0/66.5	46.5/60.6	62.1/69.1
SE_{SDLG}	73.6/70.3	81.8/**98.7**	77.4/53.8	62.1/70.7	45.6/64.6	68.1/71.6
Self-Cons.	79.0/87.8	76.8/95.8	73.8/54.4	58.3/87.2	61.8/86.6	69.9/82.4
TU	81.2/87.7	77.5/95.2	73.5/54.5	72.7/91.6	66.4/87.0	74.3/83.2
Probe-Qwen	83.8/90.2	**89.6**/98.3	71.0/53.6	**77.1/94.0**	67.2/88.7	77.7/85.0
Probe-Llama	**92.7/93.0**	76.1/95.8	**78.2/57.1**	70.0/91.4	**71.9/89.2**	**77.8/85.3**

Anchor: OLMo2-13B

Method	TriviaQA	StrategyQA	NQ	HotpotQA	CoQA	Average
	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	
P(True)	51.7/57.4	75.6/90.8	59.5/42.8	52.4/47.9	52.6/58.9	58.4/59.6
SAR	71.7/78.0	83.4/94.0	75.0/54.2	70.7/65.2	55.1/62.2	71.2/70.7
SE	63.5/70.4	83.8/94.3	71.0/51.0	66.0/60.4	53.2/59.5	67.5/67.1
SE_{SDLG}	72.5/71.0	89.8/96.8	**77.9**/52.2	75.1/64.6	52.3/63.5	73.5/69.6
Self-Cons.	75.6/88.2	86.0/94.8	67.5/51.2	78.8/79.5	66.2/83.1	74.8/79.4
TU	79.7/88.6	87.7/94.7	74.0/52.8	90.9/83.0	76.1/85.4	81.7/80.9
Probe-Qwen	78.7/90.1	**93.9/97.4**	70.9/53.9	**92.3/84.9**	81.1/**89.1**	83.4/**83.1**
Probe-Llama	**90.4/93.4**	81.8/94.4	77.7/**55.7**	84.9/79.8	**83.3**/88.6	**83.6**/82.4

Anchor: Mistral-Nemo-12B

Method	TriviaQA	StrategyQA	NQ	HotpotQA	CoQA	Average
	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	ROC/ARC	
P(True)	52.3/57.9	77.0/89.7	55.7/42.8	49.3/52.4	51.0/60.7	57.1/60.7
SAR	73.2/79.2	85.2/93.3	70.1/54.1	67.4/70.6	53.1/64.0	69.8/72.2
SE	64.2/71.1	85.6/93.5	66.4/51.0	62.2/66.0	51.6/61.3	66.0/68.6
SE_{SDLG}	73.2/71.7	91.6/96.0	73.3/52.2	71.3/70.2	50.7/65.3	72.0/71.1
Self-Cons.	79.8/90.0	87.7/94.4	69.5/53.7	66.5/85.0	40.5/77.0	68.8/80.0
TU	80.7/89.5	90.0/94.4	69.2/52.8	85.6/90.9	73.8/88.0	79.9/83.1
Probe-Qwen	81.1/90.5	**94.1/96.8**	71.0/51.9	**91.2/94.3**	75.1/**90.4**	**82.5/84.8**
Probe-Llama	**91.7/93.6**	83.0/93.4	**77.9/54.5**	78.2/88.6	**78.5**/90.0	81.9/84.0

4.2 Implementation Details

We evaluate four target models at the 12–14B scale: Qwen3-14B (Yang et al., 2025), Phi-4 (Abdin et al., 2024), OLMo-2-13B-Instruct (Team OLMo et al., 2024), and Mistral-Nemo-Instruct-2407 (Mistral AI Team, 2024). The response generated by each target model is treated as the anchor answer. We construct two scale-ordered small-model probe families. Probe-Qwen contains Qwen3-8B, Qwen3-4B, and Qwen3-1.7B, while Probe-Llama contains Llama-3.1-8B-Instruct, Llama-3.2-3B-Instruct, and Llama-3.2-1B-Instruct (Dubey et al., 2024; Yang et al., 2025). Bidirectional entailment is computed using `nli-deberta-v3-base` (He et al., 2023).

For AnchorAUC, the target model and each probe model independently generate one response per input with temperature 0. Free-text retention is measured using bidirectional entailment, while binary, multiple-choice, and numerical answers use strict normalized matching. Implementation details for all baselines are provided in Appendix B.

StrategyQA correctness is determined by strict normalized matching. For the remaining datasets, anchor answers are evaluated by DeepSeek-V4-Flash (DeepSeek-AI, 2026) with temperature 0. Strong LLM judges have been shown to align more closely with human judgments than exact-match or token-overlap metrics (Ho et al., 2025; Zheng et al., 2023; Liu et al., 2026). Further details of data generation, answer processing, and correctness-label assignment are provided in Appendix A.

5 Results and Analysis

5.1 Main Results

As shown in Table 1, the proposed probe-based configurations achieve competitive performance across anchor models and evaluation tasks. TU provides the most direct comparison because it also estimates uncertainty using multiple models. However, TU aggregates generic disagreement among models of comparable scale, whereas our method orders lightweight probes by decreasing model scale and measures their semantic retention relative to a fixed anchor answer. The resulting trajectory characterizes where disagreement emerges and how it develops along the selected probe sequence, providing a structured scale-ordered answer-retention signal.

Averaged over all anchors and tasks, Probe-Qwen achieves $80.98\%/84.49\%$ AUROC/AUARC and Probe-Llama achieves $81.34\%/84.36\%$, compared with TU's $78.10\%/82.68\%$. Probe-Llama performs particularly well on TriviaQA and NQ, whereas Probe-Qwen is stronger on StrategyQA and HotpotQA. These results indicate that scale-ordered answer retention provides a competitive and computationally efficient uncertainty signal, with complementary strengths across probe families and tasks.

Table 2: Large-anchor performance averaged over Probe-Llama and Probe-Qwen. Each entry reports AUROC/AUARC.

Anchor	Setting	TriviaQA		StrategyQA		NQ		HotpotQA		CoQA		Average	
		AUROC	AUARC	AUROC	AUARC	AUROC	AUARC	AUROC	AUARC	AUROC	AUARC	AUROC	AUARC
12–14B Anchors	TU	81.1	88.6	82.9	95.0	72.8	52.5	81.4	89.9	72.4	87.4	78.1	82.7
	RawSize	87.0	91.6	85.8	96.5	75.0	53.7	81.3	90.6	76.7	89.8	81.2	84.4
Claude Opus 5	TU	74.6	97.1	76.4	97.2	66.3	81.4	74.9	95.4	65.9	96.9	71.6	93.6
	RawSize-14B	93.7	100.0	96.6	97.8	68.8	84.0	94.6	96.0	92.0	99.8	**89.1**	**95.5**
	RawSize-100B	92.6	99.9	96.3	97.8	68.4	83.8	94.5	95.9	91.1	99.8	88.6	95.4
GPT-5.6 Sol	TU	75.1	97.5	76.9	97.4	66.8	82.4	75.4	95.7	66.4	97.4	72.1	94.1
	RawSize-14B	84.4	99.7	97.3	97.8	66.0	86.0	93.2	97.7	89.2	99.8	86.0	**96.2**
	RawSize-100B	86.9	99.7	97.0	97.8	65.9	85.9	93.2	97.7	88.6	99.8	**86.3**	**96.2**

5.2 Large-Anchor Generalization

The main experiments use 12–14B anchors, raising the question of whether anchor-relative retention remains informative when the anchor is substantially larger than the probe family. We therefore use Claude Opus 5 (Anthropic, 2026) and GPT-5.6 Sol (OpenAI, 2026) as anchors while keeping the original Probe-Qwen and Probe-Llama chains. Since their parameter counts are unavailable, we evaluate two fixed reference placements, 14B and 100B, for constructing the RawSize coordinate; these values define the reference spacing used to construct the RawSize coordinate.

As shown in Table 2, AnchorAUC remains effective when the anchor is substantially larger than the probe family. Under the 14B reference placement, Claude Opus 5 and GPT-5.6 Sol achieve average AUROC/AUARC values of 89.1/95.5 and 86.0/96.2, respectively; the corresponding results under the 100B placement are 88.6/95.4 and 86.3/96.2. This indicates that anchor-relative retention remains discriminative even under a large anchor–probe scale gap.

The advantage over TU is particularly clear in AUROC. TU obtains average AUROC/AUARC values of 71.6/93.6 for Claude Opus 5 and 72.1/94.1 for GPT-5.6 Sol, substantially below the corresponding RawSize results. This suggests that generic cross-model disagreement becomes less discriminative when the anchor is much stronger than the auxiliary models, whereas capability-ordered retention remains effective, consistent with the reported sensitivity of TU to increasing reference-model scale (Hamidieh et al., 2026).

The two RawSize placements yield very similar overall performance: Claude Opus 5 slightly favors 14B, while GPT-5.6 Sol slightly favors 100B, with nearly unchanged AUARC. The similar performance across the two placements indicates that AnchorAUC is relatively insensitive to the reference placement used for closed-source anchors.

5.3 Is the Retention Trajectory Structured?

We compare AnchorAUC with four alternative summaries constructed from exactly the same probe outputs to isolate the contribution of trajectory aggregation. Strongest-only uses the strongest-probe retention; Mean retention averages the three probe scores; Minimum retention takes their minimum; and First-failure measures the fraction of consecutive probe levels retained above 0.8 before the first failure.

Table 3: Aggregation ablation.

Aggregation	Probe-Qwen		Probe-Llama		Average	
	AUROC	AUARC	AUROC	AUARC	AUROC	AUARC
Strongest only	76.8	83.4	78.3	83.7	77.5	83.6
Mean retention	79.4	83.2	78.4	82.8	78.9	83.0
Minimum retention	74.3	82.4	67.5	80.0	70.9	81.2
First-failure	76.8	83.0	76.2	82.6	76.5	82.8
AnchorAUC	**81.0**	**84.5**	**81.3**	**84.4**	**81.2**	**84.4**

As shown in Table 3, AnchorAUC outperforms all four alternatives for both probe families. Relative to mean retention, the strongest baseline, it improves macro AUROC by 2.3 points and AUARC by 1.4 points, with paired-bootstrap 95% CIs of $[2.09, 2.38]$ and $[1.34, 1.46]$, respectively. These results show that trajectory aggregation contributes additional uncertainty information beyond the probe outputs themselves.

We next test whether the ordering of this trajectory is informative. Keeping probe answers and raw-size coordinates fixed, we enumerate all $3! = 6$ assignments of the three probes to the ordered positions.

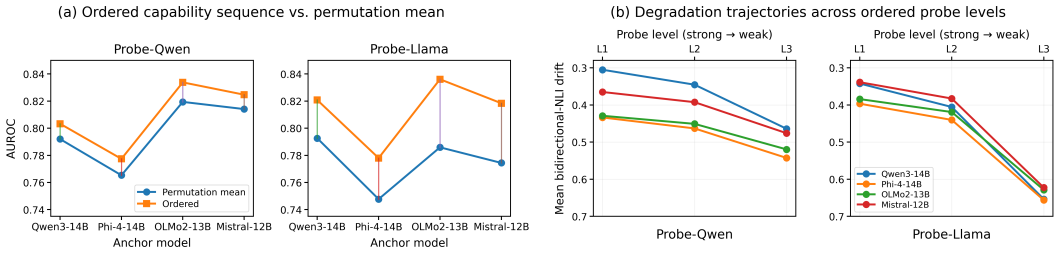


Figure 2: (a) AUROC under the scale-aligned assignment versus the mean of the five non-aligned assignments. (b) Bidirectional-NLI retention across scale-ordered probe levels. Results are macro-averaged over five tasks; full results are provided in Appendix F.

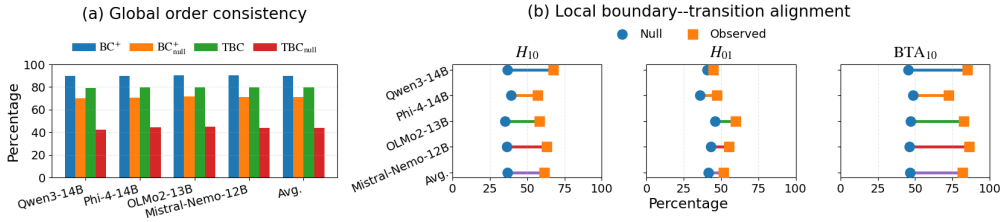


Figure 3: (a) Global correctness-order consistency against count-preserving random-order nulls. (b) Local alignment between retention-defined behavioral boundaries and correctness transitions. Results are macro-averaged over five tasks and the two probe families.

As shown in Figure 2(a), the scale-aligned assignment outperforms the mean of the five non-aligned assignments by 2.52 AUROC points on average. It ranks first in every anchor–probe configuration, whereas the fully reversed assignment ranks last (Table 9). Figure 2(b) provides complementary evidence: bidirectional-NLI retention consistently decreases with probe size across all anchors and probe families.

Together, these results show that both capability-aligned ordering and trajectory aggregation contribute useful uncertainty information. The next section examines whether independently evaluated correctness transitions follow the same ordered sequence.

5.4 Do Correctness Transitions Follow the Probe Order?

Global order consistency. Let $c_0(q) \in \{0, 1\}$ denote anchor correctness and $c_i(q)$ the correctness of the i-th probe, ordered from strongest to weakest. Let $m(q) = \sum_{i=1}^{K} c_i(q)$ and $o(q) = \mathbb{I}[c_1(q) \geq \cdots \geq c_K(q)]$. We define $\mathrm{BC}^+ = \mathbb{E}[o(q) \mid c_0(q) = 1]$ and $\mathrm{TBC} = \mathbb{E}[o(q) \mid c_0(q) = 1, m(q) < K]$, where TBC removes the all-correct ceiling pattern.

We compare both measures with an exact count-preserving random-order null. Conditional on $m(q)$, the $m(q)$ correct outcomes are uniformly assigned to the K probe positions, giving an ordered probability of $\binom{K}{m(q)}^{-1}$.

As shown in Figure 3(a), BC^+ averages 90.1%, compared with 70.9% under the null; after removing all-correct sequences, TBC remains 79.7%, versus 43.8% under the null. Thus, correctness outcomes follow the probe order substantially more often than expected from random arrangements with the same number of correct probes.

Local boundary–transition alignment. We next test whether correctness changes occur near the retention-defined boundary. Let $r_0(q) = 1$, $d_i(q) = r_{i-1}(q) - r_i(q)$, and define the first substantial retention drop as $b_\tau(q) = \min\{i : d_i(q) \geq \tau\}$, with $\tau = 0.2$. If no drop reaches the threshold, no boundary is detected.

Let $T_{10}(q)$ and $T_{01}(q)$ denote the locations of $1 \to 0$ and $0 \to 1$ correctness transitions. We use H_{10} and H_{01} for the fractions of these transitions occurring at $b_\tau(q)$, treating examples without a detected

Table 4: Anchor-matched vs. anchor-mismatched probe families.

Metric	TriviaQA		StrategyQA		NQ		HotpotQA		CoQA		Average	
	Match	Mismatch	Match	Mismatch	Match	Mismatch	Match	Mismatch	Match	Mismatch	Match	Mismatch
AUROC ↑	84.63	93.11	88.42	79.75	73.33	79.63	80.55	76.31	74.74	81.59	80.33	82.08
AUARC ↑	89.62	92.45	98.56	97.17	49.75	52.68	96.72	94.94	90.90	91.49	85.11	85.75
$\text{SER}_{0.8}$ ↓	0.90	0.50	1.20	1.70	2.20	1.50	1.90	0.60	3.50	0.60	1.94	0.98
CDR ↓	23.92	16.75	8.01	18.28	15.32	13.38	8.76	30.36	15.37	28.43	14.28	21.44

The anchor is Qwen3-14B. Match denotes Probe-Qwen and Mismatch denotes Probe-Llama. Green cells indicate the better result within each pair.

boundary as misses. To focus specifically on the onset of correctness loss, we further define

$$\text{BTA}_{10} = \Pr[b_\tau(q) = \min T_{10}(q) \mid c_0(q) = 1,\ \exists i : c_i(q) = 0]\,. \tag{7}$$

The corresponding null fixes the retention boundary and the number of correct probes while randomly permuting probe correctness over the ordered positions, using $10{,}000$ permutations.

As shown in Figure 3(b), 61.3% of $1 \to 0$ transitions occur at the detected boundary, versus 36.8% under the null, while the corresponding $0 \to 1$ rates are 51.7% and 41.3%. More directly, the first $1 \to 0$ transition aligns with the retention boundary in 81.6% of eligible examples, compared with 46.7% under the null.

Together, these results show that correctness loss tends both to follow the ordered probe sequence and to occur near the first substantial retention decline, supporting its interpretation as an observable behavioral boundary. Additional correctness-pattern diagnostics are provided in Appendix D.

5.5 Does the Probe Family Need to Match the Anchor?

Our probe chains are always constructed within a single model family. We further ask whether that family must match the anchor family. We fix Qwen3-14B as the anchor and compare the matched Probe-Qwen chain with the mismatched Probe-Llama chain.

Besides AUROC and AUARC, we consider two failure modes. Let A and P_i denote anchor and probe correctness and r_i the bidirectional-NLI retention. We define $\text{SER}_{0.8} = P(A = 0, \forall i : P_i = 0, \min_i r_i \geq 0.8)$ for highly retained shared errors, and $\text{CDR} = \frac{1}{K}\sum_{i=1}^{K} P(A = 1, P_i = 0)$ for disagreement with a correct anchor.

As shown in Table 4, matching the probe family to the anchor is not necessary: the mismatched Probe-Llama chain slightly improves average AUROC/AUARC from $80.33/85.11$ to $82.08/85.75$, although the relative advantage varies by task.

The diagnostics reveal a trade-off. The mismatched chain reduces $\text{SER}_{0.8}$ from 1.94% to 0.98%, but increases CDR from 14.28% to 21.44%. A plausible explanation is that anchor-matched probes share more family-specific biases with the anchor, preserving both correct and incorrect agreement, whereas a different-family probe chain provides less correlated evidence and is therefore more likely to expose shared errors, at the cost of additional disagreement. Despite this trade-off, shared-error rates remain low under both settings, suggesting that this failure mode is substantially mitigated. These results support using an internally consistent, ordered probe family, with flexibility in its relation to the anchor family.

5.6 More Probes Do Not Always Improve Performance

Figure 4(a)–(b) shows that increasing probe granularity is generally beneficial. Adding the intermediate probe consistently improves performance for both probe families, suggesting that a denser probe trajectory can characterize answer degradation more precisely. However, adding the weakest probe yields only a small additional gain for Probe-Qwen and reduces performance for Probe-Llama. Thus, the benefit depends on the informativeness of the added probe level rather than on probe count alone.

Figures 4(c)–(d) further reveal a clear position effect. The strongest probe contributes the most, followed by the intermediate probe, whereas the weakest probe exhibits substantial diminishing returns. The weakest Qwen probe remains slightly beneficial, while the weakest Llama probe introduces a small negative contribution. Overall, stronger and intermediate probes provide the core uncertainty signal. Additional weak probes can improve the resolution of the degradation trajectory only when they retain sufficient discriminative information; otherwise, they mainly introduce noise.

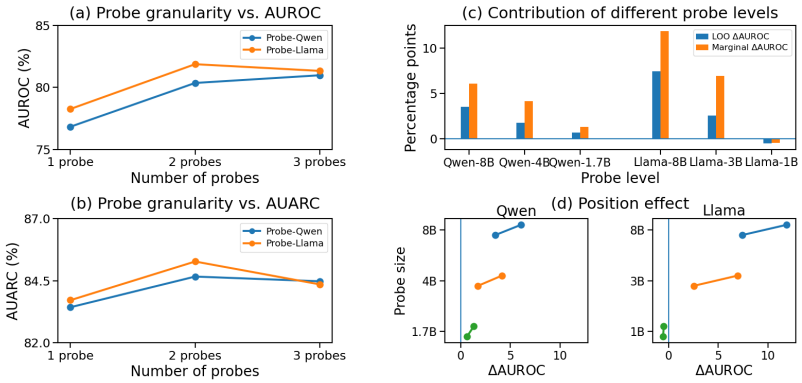
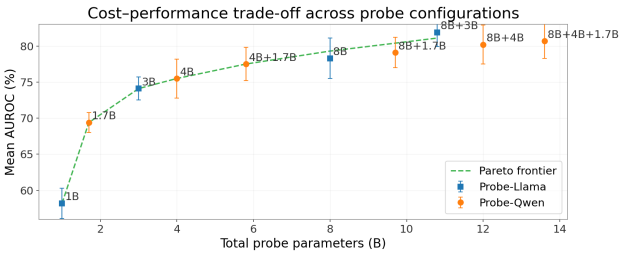


Figure 4: Probe granularity and position effects, macro-averaged over four anchor models and five evaluation tasks. (a)–(b) Performance with different numbers of probes. (c)–(d) Contributions of different probe levels. The detailed numerical results are reported in Appendix E, Tables 7 and 8.



Configuration	Calls	Param. calls (B)	Tokens	Gen. (s)	NLI (s)	Total (s)
Qwen 8B	1	8.0	656.0	3.089	0.000	3.090
Qwen 8B+4B	2	12.0	1181.9	4.931	0.001	4.931
Qwen 8B+4B+1.7B	3	13.7	1581.0	5.115	0.001	5.116
Llama 8B	1	8.0	321.7	0.744	0.001	0.745
Llama 8B+3B	2	11.0	626.7	1.032	0.001	1.033
Llama 8B+3B+1B	3	12.0	1598.5	1.193	0.002	1.195
TU	8	106.0	4380.9	39.856	–	39.856

Figure 5: Cost–performance trade-off and incremental inference cost. **Left:** Average AUROC across probe configurations; error bars denote standard deviations and the dashed line marks the Pareto frontier. **Right:** Per-example incremental inference cost of AnchorAUC and TU.

5.7 Cost–Performance Trade-off

We further examine the cost–performance trade-off under different probe subsets. All inference latency is measured on a single NVIDIA RTX PRO 6000 GPU. As shown in Figure 5, performance generally improves with stronger or additional probes, but the marginal gain decreases as the probe chain grows. Probe-Qwen 4B+1.7B provides a compact mid-cost option with an average AUROC of 77.5%, while Probe-Llama 8B+3B achieves the highest average AUROC of 81.9%. This suggests that two-probe configurations already capture most of the benefit of a longer probe chain.

AnchorAUC also shows a substantial cost advantage over TU. Probe-Llama 8B+3B requires only 1.033 seconds per example, compared with 39.856 seconds for the TU, corresponding to nearly a 39× reduction in incremental latency. Even the full Probe-Qwen chain requires only 5.116 seconds. Since bidirectional NLI contributes at most 0.002 seconds, AnchorAUC provides a clearly better cost–performance trade-off.

6 Conclusion

We proposed **AnchorAUC**, a black-box uncertainty estimation method that measures how well an anchor answer is retained across an ordered family of progressively smaller models. By summarizing the resulting retention trajectory as an area-based score, AnchorAUC requires neither model-internal probabilities nor repeated sampling from the target model. Experiments across multiple anchors, probe families, and QA benchmarks show that the method provides effective and computationally efficient uncertainty estimates. The observed semantic degradation, ordered correctness transitions, and cost–performance results further support the retention trajectory as a structured behavioral signal for black-box uncertainty estimation.

ETHICS STATEMENT

This work studies uncertainty estimation for large language models using publicly available question-answering benchmarks. All evaluated models are used only for research purposes and under their respective licenses.

Although AnchorAUC substantially reduces uncertainty-estimation failures caused by shared model errors, it cannot eliminate them entirely. As shown in Table 4, highly retained shared errors still account for approximately 1%–2% of examples across different probe-family settings. Therefore, when the anchor and probe models exhibit correlated biases or similar failure modes, AnchorAUC may still assign high confidence to a small fraction of shared errors. It should not be used as the sole basis for decisions in high-stakes domains such as healthcare, law, finance, or public safety. The use of an LLM judge may also introduce model-specific biases into correctness labels, although we mitigate this risk through deterministic matching, a unified judging protocol, caching, and label-consistency checks.

The proposed method requires additional inference from several probe models. Although the probes are smaller than the target models and require only one generation each, their computational and environmental costs should still be considered when applying the method at scale.

REPRODUCIBILITY STATEMENT

We provide the complete experimental protocol in the main paper and appendices, including the evaluated datasets and model configurations, generation prompts, decoding parameters, answer-extraction and normalization procedures, correctness-labeling protocol, retention-score computation, model-scale coordinate parameterization, and evaluation metrics.

All methods are evaluated on the same fixed set of $1{,}000$ examples per benchmark and use the same anchor answers and correctness labels. The anchor and probe models use deterministic decoding with temperature 0, whereas the sampling-based baselines follow the generation settings specified in Appendix B. The appendices further report complete probe-order, correctness-transition, granularity, coordinate, and cost–performance results.

Upon publication, we will release the implementation code, configuration files, evaluation scripts, processed example identifiers, and cached intermediate outputs permitted by the corresponding dataset and model licenses.

AI USE STATEMENT

This manuscript used a large language model only for light editorial support—namely grammar and spelling checks, minor language polishing, and table formatting. The LLM did not generate scientific content, results, analyses, or claims. All edits were reviewed by the authors, and the authors remain fully responsible for the final text.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 Technical Report. *arXiv preprint arXiv:2412.08905*, 2024. doi: 10.48550/arXiv.2412.08905.

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. Improving uncertainty estimation through semantically diverse language generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b94d8b035e2183e47afef9e2f299ba47-Abstract-Conference.html.

Anthropic. Introducing Claude Opus 5. https://www.anthropic.com/news/claude-opus-5, July 2026. Published July 24, 2026. Accessed August 6, 2026.

Jasper Meynard Arana, Kristine Ann M. Carandang, Ethan Robert Casin, Christian Alis, and Christopher Monterola. When models hesitate: Answer instability as a label-free uncertainty signal for LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 816–826, San Diego, California, United States, jul 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-srw.72. URL https://aclanthology.org/2026.acl-srw.72/.

Jiuhai Chen and Jonas Mueller. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5186–5200, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.283. URL https://aclanthology.org/2024.acl-long.283/.

Huizi Cui, Huan Ma, Qilin Wang, Yuhang Gao, and Changqing Zhang. Estimating the black-box LLM uncertainty with distribution-aligned adversarial distillation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42721–42740, San Diego, California, United States, jul 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1979. URL https://aclanthology.org/2026.acl-long.1979/.

DeepSeek-AI. DeepSeek-V4-Flash. https://huggingface.co/deepseek-ai/DeepSeek-V4-Flash, 2026. Model card. Accessed: 2026-07-12.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5050–5063, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.276. URL https://aclanthology.org/2024.acl-long.276/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Kimia Hamidieh, Veronika Thost, Walter Gerych, Mikhail Yurochkin, and Marzyeh Ghassemi. Complementing self-consistency with cross-model disagreement for uncertainty quantification. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lOoRJo8xWy.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Xanh Ho, Jiahao Huang, Florian Boudin, and Akiko Aizawa. LLM-as-a-Judge: Reassessing the performance of LLMs in extractive QA. *arXiv preprint arXiv:2504.11972*, 2025. doi: 10.48550/arXiv.2504.11972.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2), 2025. doi: 10.1145/3703155.

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, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

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, 2022. URL https://arxiv.org/abs/2207.05221.

Adam Tauman Kalai and Santosh S. Vempala. Calibrated language models must hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pp. 160–171. Association for Computing Machinery, 2024. doi: 10.1145/3618260.3649777. URL https://doi.org/10.1145/3618260.3649777.

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, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=DWkJCSxKU5.

Hongye Liu, Dhanajit Brahma, and Ricardo Henao. Calibrating model-based evaluation metrics for summarization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35285–35315, 2026.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Mistral AI Team. Mistral NeMo. Mistral AI, July 2024. URL https://mistral.ai/news/mistral-nemo/. Accessed: 2026-08-04.

Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. Accuracy-rejection curves (ARCs) for comparing classification methods with a reject option. In *Proceedings of the Third International Workshop on Machine Learning in Systems Biology*, volume 8 of *Proceedings of Machine Learning Research*, pp. 65–81. PMLR, 2009. URL https://proceedings.mlr.press/v8/nadeem10a.html.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929, 2024.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, July 2026. Published July 9, 2026. Accessed August 6, 2026.

Xin Qiu and Risto Miikkulainen. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. *Advances in neural information processing systems*, 37:134507–134533, 2024.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019. doi: 10.1162/tacl_a_00266. URL https://aclanthology.org/Q19-1016/.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 OLMo 2 Furious. *arXiv preprint arXiv:2501.00656*, 2024. doi: 10.48550/arXiv.2501.00656.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Data Generation and Processing

A.1 Input Representation and Canonicalization

Each evaluation instance consists of a question q, an optional context c, and one or more reference answers $\mathcal{G}$. Before answer generation, we canonicalize the input fields to avoid duplicating the evidence context. If the question field already contains an explicit context block, the separate context field is cleared; otherwise, the question and context are retained as separate fields. The original question, context, reference answers, and example identifier are preserved throughout answer generation, correctness labeling, and subsequent evaluation.

A.2 Target and Probe Answer Generation

The target model and all probe models independently answer the same question–context pair. The anchor answer generated by the target model is never included in the input to any probe model.

We use a unified structured generation format for the target and probe models. Each model is instructed to place its reasoning inside `[THOUGHT]` and `[/THOUGHT]` tags and its concise final answer inside `[FINAL ANSWER]` and `[/FINAL ANSWER]` tags. Generation uses deterministic decoding with `temperature` $= 0$.

A.3 ANSWER-GENERATION PROMPT

The target and probe models use the same system instruction:

```
Follow the requested output format exactly.
Think step by step, and put the complete reasoning process
only inside [THOUGHT] and [/THOUGHT].
Put only the final concise short answer inside [FINAL
ANSWER] and [/FINAL ANSWER].
The final-answer field must contain no explanation,
justification, preamble, or full sentence when an entity,
number, date, place, option, or Yes/No is sufficient.  Never
omit either section.  Do not output text outside the tags.
```

The system instruction is followed by the following one-shot formatting demonstration:

```
User:
Question:
Which planet is commonly called the Red Planet?
Assistant:
[THOUGHT]
Think step by step.  The nickname refers to the planet whose
iron-rich surface appears reddish.
[/THOUGHT]
[FINAL ANSWER]
Mars
[/FINAL ANSWER]
```

For each evaluation instance, the user prompt is:

```
Solve the QA input independently.
Think step by step.  Write the reasoning process inside
[THOUGHT]. Then write only the final concise short answer
inside [FINAL ANSWER].
Return exactly:
[THOUGHT]
Think step by step using only the facts needed to answer the
question.  Use at most 3 concise reasoning steps and at most
80 words.
[/THOUGHT]
[FINAL ANSWER]
Only the final short answer.  No explanation, no preamble,
and no complete sentence when a short entity, number, date,
place, option, or Yes/No is sufficient.  Use at most 10
words.
[/FINAL ANSWER]
All four tags are mandatory.  Do not output text outside the
tags.
Context:
{context}
Question:
{question}
```

When no context is available, the `Context` block is omitted. If the generated response cannot be parsed, the following instruction is appended to the original prompt before retrying:

```
STRICT RETRY: Think step by step and use at most 50
words inside [THOUGHT]. Then include a nonempty [FINAL
ANSWER] containing only the concise short answer, with
no explanation.  All four opening and closing tags are
mandatory.
```

A.4 Final-Answer Extraction and Cleaning

For every structured model output, the parser first searches for complete `[THOUGHT]` and `[FINAL ANSWER]` sections. It also accepts minor formatting variants, including angle-bracket tags, `[ANSWER]`, `[FINAL]`, and headings such as `Answer:`, `Final Answer:`, or `Reasoning:`.

If a response cannot be parsed, the model is queried again using the strict retry instruction described above. Examples that remain unparseable after all retries are excluded and recorded in a separate skipped-example file. For successfully parsed examples, the reasoning and final answer are stored separately, while all subsequent experiments use only the final-answer field.

During cleaning, residual reasoning and answer tags, including `<think>`, `<final>`, and `<final_answer>`, are removed. Common answer prefixes such as `Answer:`, `Final Answer:`, and `Conclusion:` are also removed. If a response contains a construction such as "the answer is X," only X is retained. Common leaked reasoning prefixes, such as `Wait`, `Let me`, `We need`, and `The question asks`, are discarded. Abnormally long cleaned answers are truncated before subsequent processing.

A.5 Generation Auditing

All generated answers are audited before correctness labeling. We first verify that each output file preserves the source row count and row order and contains all required answer fields. We then check each model output for empty responses, API error messages, timeouts, connection failures, and service errors.

The audit additionally detects residual reasoning tags, prompt echoes, refusal expressions, and abnormally long responses. For each model and task, we record the number of unique normalized answers, the answer-length distribution, and the frequency of the most common answer to identify degenerate or excessively repetitive generation.

For StrategyQA, every answer must be deterministically parseable as `Yes` or `No`. We also sample a fixed set of examples from each task and export their questions, reference answers, anchor answers, and probe answers for manual inspection.

A.6 Uniform Correctness Labeling

The target and probe answers are labeled using the same deterministic-first correctness protocol.

For StrategyQA, each prediction and reference answer is normalized to `yes` or `no` and compared directly. Only an explicit leading polarity expression is used; polarity is not inferred from an arbitrary declarative sentence.

For the remaining tasks, the prediction and all reference aliases are first normalized. The normalization procedure applies Unicode NFKC normalization, lowercasing, removal of residual reasoning and answer tags, removal of common answer prefixes and English articles, normalization of numerical separators, punctuation removal, and whitespace collapsing. If the normalized prediction exactly matches any normalized reference answer, it is labeled correct without invoking the LLM judge.

All unresolved cases are evaluated by an LLM judge. The judge receives the question, optional context, complete set of reference answers, and model prediction. It returns `CORRECT` only when the prediction expresses the same complete answer meaning as the reference answers. Incorrect entities, values, dates, numbers, units, polarity, relations, argument direction, scope, qualifiers, incomplete answers, and materially false additions are labeled `INCORRECT`. When context is available, it is treated as the source of truth.

The model-specific correctness label is stored as `correct__<model-id>`. The corresponding `correct_mode__<model-id>` field records whether the label was obtained through binary matching, normalized exact matching, judge-cache reuse, or an LLM-judge call. Judge results are cached using the task, question, context, reference answers, and prediction.

A.7 LLM-JUDGE PROMPT

For examples that cannot be resolved through deterministic matching, the following prompt is sent to the correctness judge:

```
Judge whether Prediction has the same complete answer
meaning as the Reference answer(s) for the Question.
Rules:
1. CORRECT only for the same complete answer.
2. Accept aliases, spelling variants, abbreviations, and
harmless concise paraphrases.
3. Reject wrong entity, value, date, number, unit,
polarity, relation, direction, scope, qualifier, incomplete
answers, or false additions.
4. Use the supplied context as the source of truth and
judge exactly as written.
Return exactly one word: CORRECT or INCORRECT.
Question:
{question}
Context:
{context}
Reference answer(s):
{reference_answers}
Prediction:
{prediction}
```

For examples without an input context, the `Context` block is omitted. The judge uses deterministic decoding with `temperature` $= 0$. The response parser examines both the visible response content and, when present, the `reasoning_content` field. The final standalone occurrence of `CORRECT` or `INCORRECT` is used as the verdict. If the response cannot be parsed, the judge call is repeated according to the predefined retry policy.

A.8 LABEL-CONSISTENCY REPAIR

After uniform relabeling, model answers are grouped within each example according to their normalized answer text. Identical normalized answers are required to receive identical correctness labels.

If an identical-answer group contains the anchor answer, the target-model label is treated as authoritative and propagated to the matching probe answers. This avoids assigning different labels to exactly the same answer through separate judge calls. If the group does not contain the anchor answer, it is first reevaluated using deterministic StrategyQA matching or normalized exact matching. Groups that remain unresolved are assigned a common label using a new LLM-judge call.

After repair, the target-model label is synchronized to the general `label`, `correct`, and `llm_correct_binary` fields. A processed dataset is accepted for evaluation only when all of the following conditions hold:

- the number of failed judge calls is zero;
- the number of missing correctness labels is zero;
- the number of remaining identical-answer label conflicts is zero.

The original result files are backed up before repaired labels are written, ensuring that the complete processing procedure remains traceable.

B BASELINE IMPLEMENTATION DETAILS

For P(True), we follow the protocol of Kadavath et al. (2022) to obtain self-evaluated confidence scores.

For SAR, SE, $\mathrm{SE}_{\mathrm{SDLG}}$, and Self-Cons., the target model independently generates five additional responses per input with temperature 1.0. $\mathrm{SE}_{\mathrm{SDLG}}$ uses the same response-generation protocol and base implementation settings as SE.

For TU, all four target models independently generate two additional responses per input with temperature 1.0. For a given anchor model, its two stochastic responses provide within-model variation, while the responses from the other three models serve as cross-model auxiliary evidence.

C Model-Scale Coordinate Analysis

AnchorAUC assigns a one-dimensional coordinate to each model level in the ordered anchor–probe sequence. With the generated answers and anchor-relative retention scores fixed, different coordinate parameterizations change the horizontal distances between adjacent model levels and therefore their contributions to the area under the retention curve. We compare logarithmic and raw model-scale coordinates to examine how the representation of scale gaps affects uncertainty estimation.

LogSizeAUC. The logarithmic-size scheme maps model scales onto the normalized coordinate

$$z_i^{\log} = \frac{\log \bar{s}_0 - \log s_i}{\log \bar{s}_0 - \log s_K}, \qquad i = 0, \ldots, K, \tag{8}$$

where $\bar{s}_0$ denotes the scale assigned to the anchor position. This parameterization compresses large model-scale gaps before constructing the retention curve.

RawSizeAUC. The raw-size scheme instead uses

$$z_i^{\mathrm{raw}} = \frac{\bar{s}_0 - s_i}{\bar{s}_0 - s_K}, \qquad i = 0, \ldots, K, \tag{9}$$

which preserves the relative raw-scale gaps between adjacent model levels. Parameter count is used only as a heuristic spacing coordinate for the selected degradation sequence, rather than as a calibrated numerical measure of model capability.

For either coordinate system, the retention scores are aggregated using trapezoidal integration:

$$A_{\mathcal{Z}}(x) = \sum_{i=0}^{K-1} \left(z_{i+1}^{\mathcal{Z}} - z_i^{\mathcal{Z}}\right) \frac{r_i + r_{i+1}}{2}, \qquad \mathcal{Z} \in \{\log, \mathrm{raw}\}. \tag{10}$$

Table 5 reports the results for four anchor models and two probe families. Probe-Qwen consists of Qwen3-8B, Qwen3-4B, and Qwen3-1.7B, whereas Probe-Llama consists of Llama-3.1-8B, Llama-3.2-3B, and Llama-3.2-1B.

RawSizeAUC consistently outperforms LogSizeAUC across all eight anchor–probe configurations. On average, RawSizeAUC improves AUROC from 0.8044 to 0.8116, a gain of 0.0072, while improving AUARC by 0.0027 and reducing AURC by the same amount. These results suggest that preserving raw relative scale gaps provides more useful spacing information than logarithmically compressing them. We therefore use RawSizeAUC as the default coordinate parameterization when an anchor reference scale is available.

D Correctness-Pattern Diagnostics

Table 6 reports the four non-increasing correct-anchor patterns, the remaining anchor-correct non-monotone patterns, and the all-incorrect pattern. The 1111, 1110, 1100, 1000, and non-monotone frequencies are conditioned on $c_0 = 1$. Across all configurations, anchor-correct non-monotone patterns account for only 9.2% to 11.0% of examples. Thus, approximately 89% to 91% of correct-anchor examples follow one of the four non-increasing patterns.

Table 5: Comparison of model-scale coordinates across four anchor models and two probe families. Higher AUROC and AUARC and lower AURC indicate better performance. Bold values denote the better result within each configuration.

Anchor model	Probe family	LogSizeAUC			RawSizeAUC		
		AUROC	AUARC	AURC	AUROC	AUARC	AURC
Qwen3-14B	Probe-Qwen	0.8011	0.8497	0.1503	**0.8033**	**0.8511**	**0.1489**
	Probe-Llama	0.8110	0.8543	0.1457	**0.8208**	**0.8575**	**0.1425**
Phi-4-14B	Probe-Qwen	0.7731	0.8473	0.1527	**0.7774**	**0.8494**	**0.1506**
	Probe-Llama	0.7724	0.8513	0.1487	**0.7780**	**0.8528**	**0.1472**
OLMo-2-13B	Probe-Qwen	0.8291	0.8283	0.1717	**0.8338**	**0.8307**	**0.1693**
	Probe-Llama	0.8217	0.8184	0.1816	**0.8361**	**0.8238**	**0.1762**
Mistral-Nemo-12B	Probe-Qwen	0.8205	0.8458	0.1542	**0.8247**	**0.8479**	**0.1521**
	Probe-Llama	0.8066	0.8365	0.1635	**0.8184**	**0.8400**	**0.1600**
Macro average		0.8044	0.8415	0.1586	**0.8116**	**0.8442**	**0.1559**

Table 6: Correctness-pattern frequencies along the ordered anchor–probe sequence. Patterns are written as $c_0c_1c_2c_3$.

Anchor	Probe	1111	1110	1100	1000	Non-monotone	0000 overall	0000 $\mid c_0 = 0$
Qwen3-14B	Probe-Qwen	57.00	16.80	7.69	9.00	9.51	20.62	59.95
Qwen3-14B	Probe-Llama	36.13	35.39	9.88	7.60	11.00	15.24	47.63
Phi-4-14B	Probe-Qwen	55.40	15.31	7.50	12.41	9.38	18.52	52.44
Phi-4-14B	Probe-Llama	35.99	35.24	9.86	8.25	10.65	14.98	46.64
OLMo2-13B	Probe-Qwen	55.91	14.43	7.13	13.33	9.20	18.30	44.33
OLMo2-13B	Probe-Llama	37.95	35.91	8.62	7.55	9.97	15.96	39.00
Mistral-Nemo-12B	Probe-Qwen	56.72	15.12	7.01	11.76	9.38	18.94	49.48
Mistral-Nemo-12B	Probe-Llama	38.19	35.79	8.80	7.10	10.11	15.94	43.68

The 1111, 1110, 1100, 1000, and non-monotone columns are conditioned on $c_0 = 1$ and sum to 100% within each configuration. The 0000 overall column uses all examples as the denominator, whereas 0000 $\mid c_0 = 0$ is conditioned on an incorrect anchor. All values are percentages and are macro-averaged over five tasks.

Probe-Qwen produces more all-correct 1111 sequences, whereas Probe-Llama produces substantially more 1110 transitions. This difference explains why Probe-Qwen obtains slightly higher BC^+, while Probe-Llama obtains higher TBC and larger gains over the count-preserving null.

The all-incorrect 0000 pattern accounts for 15.0% to 20.6% of all examples and for 39.0% to 60.0% of incorrect-anchor examples. These shared failures cannot inflate BC^+ or TBC because both measures are conditioned on a correct anchor.

E Detailed Probe Granularity and Position Results

Tables 7 and 8 provide the detailed numerical results underlying Figure 4.

Table 7: Performance with different probe granularities. All values are percentages and are macro-averaged over five evaluation tasks.

Anchor	Probe family	1 Probe		2 Probes		3 Probes	
		AUROC	AUARC	AUROC	AUARC	AUROC	AUARC
Qwen3-14B	Probe-Qwen	74.48	84.64	78.92	85.46	80.33	85.11
Phi-4-14B	Probe-Qwen	74.38	83.66	77.32	84.99	77.74	84.94
OLMo2-13B	Probe-Qwen	79.98	81.80	83.09	83.21	83.38	83.07
Mistral-Nemo-12B	Probe-Qwen	78.47	83.62	82.10	85.01	82.47	84.79
Qwen3-14B	Probe-Llama	78.52	85.54	82.41	86.70	82.08	85.75
Phi-4-14B	Probe-Llama	74.48	83.89	78.07	85.80	77.80	85.28
OLMo2-13B	Probe-Llama	81.15	81.81	84.49	83.50	83.61	82.38
Mistral-Nemo-12B	Probe-Llama	78.89	83.62	82.52	85.10	81.84	84.00

The one-probe setting uses the strongest 8B probe; the two-probe setting adds the intermediate probe; and the three-probe setting uses the full ordered probe sequence. Green cells indicate the best result within each anchor–family group.

Table 8: Contributions of different probe levels. Values are percentage-point changes in AUROC averaged over four anchor models.

Probe family	Position	Probe	LOO ΔAUROC	Marginal ΔAUROC
Probe-Qwen	Strong	8B	+3.49	+6.05
	Middle	4B	+1.74	+4.14
	Weak	1.7B	+0.63	+1.31
Probe-Llama	Strong	8B	+7.44	+11.86
	Middle	3B	+2.56	+6.94
	Weak	1B	−0.54	−0.47

LOO denotes the AUROC of the full ordered probe sequence minus the AUROC obtained after removing the corresponding probe. Marginal contribution is the average AUROC gain from adding that probe across all eligible subsets.

F Detailed Probe-Order Validation Results

Tables 9 and 10 report the detailed results underlying the probe-order assignment and semantic-drift analyses in Figure 2. For each probe family, indices 1, 2, and 3 denote the strongest, intermediate, and weakest probes according to model scale, respectively. Thus, 123 is the scale-aligned assignment. For Probe-Qwen, the three indices correspond to Qwen3-8B, Qwen3-4B, and Qwen3-1.7B. For Probe-Llama, they correspond to Llama-3.1-8B, Llama-3.2-3B, and Llama-3.2-1B.

Table 9: AUROC under all six assignments of the three probe models to the raw-size model-scale coordinates. Indices 1, 2, and 3 denote the strongest, intermediate, and weakest probes, respectively. The 123 assignment is scale-aligned. Alt. mean is the average over the five non-aligned assignments, and Δ is the scale-aligned AUROC minus Alt. mean. All values except rank are reported as percentages. Bold values denote the best assignment, and underlined values denote the best non-aligned assignment.

Anchor model	Probe family	123	132	213	231	312	321	Alt. mean	Δ	Rank
Qwen3-14B	Probe-Qwen	**80.33**	80.06	79.86	79.02	78.86	78.21	79.20	+1.13	1/6
Qwen3-14B	Probe-Llama	**82.08**	81.16	81.19	78.88	78.12	76.93	79.26	+2.82	1/6
Phi-4-14B	Probe-Qwen	**77.74**	77.43	77.21	76.49	76.05	75.48	76.53	+1.21	1/6
Phi-4-14B	Probe-Llama	**77.80**	76.83	77.15	74.64	72.92	72.32	74.77	+3.03	1/6
OLMo2-13B	Probe-Qwen	**83.38**	82.77	83.00	82.03	81.19	80.74	81.94	+1.44	1/6
OLMo2-13B	Probe-Llama	**83.61**	82.16	82.46	78.19	75.74	74.37	78.59	+5.03	1/6
Mistral-Nemo-12B	Probe-Qwen	**82.47**	81.85	82.26	81.19	81.13	80.64	81.41	+1.06	1/6
Mistral-Nemo-12B	Probe-Llama	**81.84**	80.13	80.94	76.90	75.16	74.04	77.44	+4.41	1/6

Table 10: Mean bidirectional-NLI drift across scale-ordered probe levels. L1, L2, and L3 denote the strongest, intermediate, and weakest probes, respectively. All drift values and slopes are reported as percentages.

Anchor model	Probe family	L1 drift	L2 drift	L3 drift	Slope
Qwen3-14B	Probe-Qwen	30.52	34.54	46.42	+28.96
Qwen3-14B	Probe-Llama	34.23	40.55	65.38	+49.89
Phi-4-14B	Probe-Qwen	43.37	46.32	54.27	+19.92
Phi-4-14B	Probe-Llama	39.66	44.03	65.63	+41.14
OLMo2-13B	Probe-Qwen	42.91	45.09	51.97	+15.12
OLMo2-13B	Probe-Llama	38.42	41.91	62.88	+35.47
Mistral-Nemo-12B	Probe-Qwen	36.51	39.25	47.61	+16.91
Mistral-Nemo-12B	Probe-Llama	33.87	38.27	62.26	+37.89