Uncovering Confident Failures: The Complementary Roles of Aleatoric and Epistemic Uncertainty in LLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) often produce confident yet incorrect responses, and uncertainty quantification in LLMs is one potential solution to more robust usage. Recent works routinely rely on self-consistency to estimate aleatoric uncertainty (AU), yet this proxy collapses precisely when models are overconfident, and produce the same incorrect answer across samples. We address this failure mode by introducing an epistemic term that measures semantic disagreement across a small ensemble of scale-matched LLMs. Specifically, we operationalize epistemic uncertainty (EU) as the gap between inter-model and intra-model response similarity, and define total uncertainty (TU) as the sum of AU and EU. The estimator is training-free and uses only black-box outputs: a few responses per model suffice. Across a range of LLMs, and long-form generation tasks, we compare TU to AU and measure uncertainty calibration by AUROC with respect to correctness and selective abstention via uncertainty thresholding. We find that TU consistently achieves higher AUROC in predicting correctness and improves selective abstention compared to AU alone. EU further exposes confident errors that AU misses, especially on tasks with near-unique correct answers, and improves the reliability of LLM uncertainty estimates.

1 Introduction

Reliable uncertainty estimates are a prerequisite for deploying large language models (LLMs) in high-stakes domains [6]. Many existing approaches for LLM uncertainty estimation are based on model's *self*-confidence [60, 57, 51], such as by measuring response consistency under sampling [27, 35, 45, 3] or querying for a verbalized uncertainty score [34]. These metrics capture how internally confident a model is in its prediction – a notion of *predictive aleatoric uncertainty* (AU). But this leaves an important question unanswered: how confident should *we* be in the model? A model might be *confident but wrong*, such as responding with the same incorrect answer with high probability (see Figure 1). In these cases, methods that rely on self-consistency can fail [26]. To address this, we focus on estimating *epistemic uncertainty* (EU) – uncertainty in our *choice* of model – which better reflects whether a model's confidence is trustworthy for a given input.

Estimating EU requires evaluating a distribution of plausible models, which is prohibitively costly for LLMs, as training even one additional model adds significant overhead [25, 9]. Recent shortcuts approximate EU in logit space [39], inject Bayesian noise during decoding [37, 15], or rely on verifier–model disagreement [61], but each imposes strong task or architecture-specific assumptions. We instead capitalize on the ecosystem of open-weight LLMs: sampling responses from a small, scale-matched ensemble lets us estimate EU directly from cross-model semantic disagreement, without additional training. While prior work has shown that LLM ensembles can improve accuracy [38, 11, 7, 50, 21], their use for uncertainty quantification has not been systematically explored.

By enabling scalable estimation of EU from model outputs alone, we can combine it with AU to obtain a more robust measure of uncertainty: Total Uncertainty (TU). These two forms of uncertainty are

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

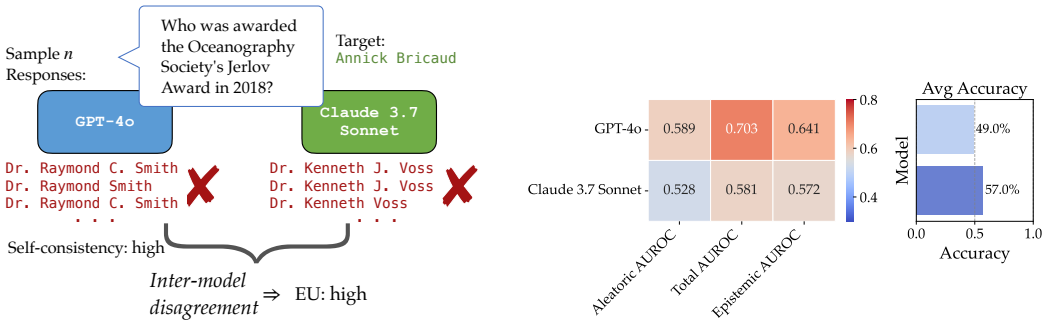


Figure 1: (a) Two models confidently produce distinct, incorrect answers to a factual question, which results in low intra-model variability (AU) but high semantic disagreement across models (EU). (b) Total uncertainty (TU = AU + EU) effectively improves uncertainty calibration with correcness in terms of AUROC on `SimpleQA`.

complementary; AU reflects variability in a model's own predictions, while EU measures divergence from other plausible models [49]. Together, they allow TU to account for both internal inconsistency and external disagreement. We estimate TU by computing semantic similarity between a reference model's responses, and those from an ensemble of LLMs. Specifically, we (1) sample responses from a set of given models, (2) compute pairwise semantic similarities to quantify both intra-model (aleatoric) and inter-model (epistemic) disagreement, and (3) define TU as the combination of aleatoric and epistemic components derived from these response distributions.

We evaluate EU, AU and TU on two standard axes: ranking-based calibration (via AUROC) and selective prediction (via abstention under uncertainty thresholds), across a range of models and generation tasks [27, 46]. We conduct experiments across five 7–9B parameter Instruction-tuned models [62, 22, 17, 16], on ten long-form generation tasks spanning QA, summarization, translation, and math reasoning [24, 63, 42]. We also repeat these experiments for API models such as `GPT-4o` [20] and `Claude 3.7 Sonnet` [1] on `SimpleQA` [59].

Our findings are as follows:

- TU consistently outperforms AU in both AUROC and selective abstention. For example, TU improves uncertainty calibration from 0.59 to 0.70 in AUROC on `SimpleQA` for `GPT-4o`.
- EU reliably identifies *confident but incorrect generation*, i.e., cases where aleatoric uncertainty is low but the model is wrong.
- Our notion of EU is most informative in tasks with a single correct answer, such as factual QA (`HotpotQA`) and translation (`WMT16`).

2 Related Works

Aleatoric Uncertainty in LLMs Existing approaches predominantly focus on AU, which captures response inconsistency or input ambiguity. Recent surveys provide extensive reviews of these methods [60, 51, 57]. Typical strategies involve sampling multiple responses per prompt and analyzing their semantic consistency, often through clustering or entropy-based metrics [35].

Bayesian-inspired EU Estimation. A line of research employs Bayesian-inspired methods, such as adding noise to embeddings during generation to approximate uncertainty in model weights [37], sampling from a model with different temperatures [15], or leverage entropy from decoding from different hidden states as proxy for uncertainty, which provides a computationally efficient alternative to exhaustive sampling [14]. Training ensembles explicitly, such as LoRA-based methods [58], demonstrate improved uncertainty calibration but incur significant computational costs. Another work [39] calculates AU and EU on token level by considering the LLM logits as parameters of a Dirichlet distribution and by applying other UQ methods subsequently. Such token-level scores can complement sequence-level scores as studied by us.

Prompt-Based and Verifier-Based EU Estimation. Prior works in iterative prompting [2, 23] estimate epistemic uncertainty by iteratively querying the same model, adding previous responses to the later queries' prompts, and measuring probabilistic inconsistencies as indicators of hallucination. However, these methods have limitations: gains over AU are mainly observed on multi-label data with limited benefits shown in standard single-label QA [2]. Others only experiment on synthetic data [23].

Another work utilizes one verifier LLM and shows that inter-model disagreement as a proxy for EU complements AU in cases where we reach the performance bounds of self-consistency [61]. While they also experiment with weaker verifiers, our study offers deeper insight into the interplay between different models of different capability. Moreover, in their practical proposal, they suggest to consider cross-consistency only if AU is of intermediate range, while our evaluation and also [52] suggest that especially low AU is prone to hallucination and well complemented by EU (see Sec. 5.1). Prior works are mostly limited to special kinds of question-answering data, do not account for the impact of the vast LLM model space on EU, and do not fully explore interactions between AU and EU, gaps our work addresses explicitly.

LLM Ensembles. Our work builds upon classical uncertainty estimation, such as deep and dropout ensembles [31, 13] and is closely related to LLM ensemble applications [38]. In particular, various recent works focus on LLM collaborations [11], verifier LLMs [32], and sampling from multiple LLMs [7]. We study LLM ensembles from the viewpoint of uncertainty estimation.

3 Quantifying Predictive Uncertainty Using Response Similarity

Let ω be the particular parameterization of an LLM, and let x be a prompt. Our goal is to quantify the predictive uncertainty of ω given x as input. As is standard, we categorize predictive uncertainty into two sub-components: AU and EU [19]. The aleatoric component captures the inherent unpredictability of the response to x under the model ω, while the epistemic component captures our uncertainty in ω being the correct parametrization to use when responding to input x. We define total predictive uncertainty additively as the sum of the aleatoric and epistemic uncertainties.

3.1 Aleatoric Uncertainty via Intra-Model Response Similarity

Many recent works have proposed techniques to measure the randomness in LLM responses [28, 35, 36]. These techniques typically focus on measures of *semantic* uncertainty, where uncertainty is defined as a function of how often an LLM produces semantically distinct outputs given the same input [28, 35]. In particular, Lin *et al.* [35] propose a measure equivalent to[1] the following:

$$U_{\text{aleatoric}}(x;\omega) = \mathbb{E}_{r_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{r_2^\omega \sim p(\cdot|x,\omega)} \left[1 - s(r_1^\omega, r_2^\omega)\right], \tag{1}$$

where $s(\cdot,\cdot)$ is a similarity metric for responses such as the cosine similarity in an embedding space.

In essence, Equation 1 corresponds to the expected similarity between two responses independently sampled from $p(\cdot|x,\omega)$, the response distribution of ω conditioned on x. If responses typically have the same semantic meaning as each other, meaning that the meaning of the response does not vary when resampled, then $U_{\text{aleatoric}}(x;\omega)$ will be close to 0, which means that there is little uncertainty in how ω will respond to x. If the model is likely to produce semantically distinct responses for the same input, then $U_{\text{aleatoric}}(x;\omega)$ will be high, which means ω has high uncertainty for x.

Equation 1 captures the inherent uncertainty in the response to x given model ω. However, ω may not be the optimal model to use for x, and Equation 1 fails to capture the inherent uncertainty that comes from choosing ω as our parameterization. There is thus a need to also capture the *epistemic* uncertainty that comes from our model choice.

3.2 Epistemic Uncertainty as Inter-Model Response Similarity

Let ω^* represent a hypothetical "ideal" model, such that $p(\cdot|x;\omega^*) = p(\cdot|x)$; the distribution of responses from ω^* equals the true response distribution. We can thus quantify the epistemic uncertainty of ω as a divergence between ω and ω^*; e.g., $U_{\text{epistemic}}(x,\omega) = D(\omega \mid\mid \omega^*)$ [49]. We define D as follows:

$$D(\omega \mid\mid \omega^*) = -\Big[\underbrace{\mathbb{E}_{q_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{q_2^{\omega^*} \sim p(\cdot|x,\omega^*)} \left[s(q_1^\omega, q_2^{\omega^*})\right]}_{\text{cross-model similarity}} - \underbrace{\mathbb{E}_{r_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{r_2^\omega \sim p(\cdot|x,\omega)} \left[s(r_1^\omega, r_2^\omega)\right]}_{\text{self-similarity (1 - AU)}}\Big]. \tag{2}$$

In effect, $D(\omega \mid\mid \omega^*)$ measures the difference between 1) the similarity of responses from ω and ω^* $\left(\mathbb{E}_{q_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{q_2^{\omega^*} \sim p(\cdot|x,\omega^*)} \left[s(q_1^\omega, q_2^{\omega^*})\right]\right)$ and 2) the self-similarity of ω

[1] This is equivalent to U_{Deg} in [35].

$\left(\mathbb{E}_{r_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{r_2^\omega \sim p(\cdot|x,\omega)} \left[s(r_1^\omega, r_2^\omega) \right] \right)$. In the case where ω is optimal and equivalent to ω^*, then $D(\omega \,||\, \omega^*)$ will be 0. When ω produces responses that are semantically diverse from the ideal model's responses even after accounting for the diversity due to ω's aleatoric uncertainty, then $D(\omega \,||\, \omega^*)$ will be high.

In practice, we do not have access to the optimal model ω^*. Instead, we can leverage a recent information-theoretic technique [49] and marginalize out ω^*. Let P_Ω be a distribution over models such that $\mathbb{E}_{\tilde{\omega} \sim P_\Omega} \left[p(\cdot \mid x; \tilde{\omega}) \right] = p(\cdot \mid x)$. We can thus replace ω^* in Equation 2 with an expectation over P_Ω, and define $U_{\text{epistemic}}(x, \omega)$ as:

$$\begin{aligned} U_{\text{epistemic}}(x, \omega) = &- \mathbb{E}_{\tilde{\omega} \sim P_\Omega} \left[\mathbb{E}_{q_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{q_2^{\tilde{\omega}} \sim p(\cdot|x,\tilde{\omega})} \left[s(q_1^\omega, q_2^{\tilde{\omega}}) \right] \right] \\ &+ \mathbb{E}_{r_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{r_2^\omega \sim p(\cdot|x,\omega)} \left[s(r_1^\omega, r_2^\omega) \right]. \end{aligned} \tag{3}$$

When the average similarity in responses between ω and other sampled models matches the self-similarity of ω's responses, then the semantic distribution of ω matches the target distribution and the epistemic uncertainty is low. When there is a mismatch between the average similarity of ω's responses to the responses from the sampled models $\tilde{\omega}$ compared to ω's self-similarity, then there is a disagreement in how models respond and the epistemic uncertainty is high. In Appendix A.1, we provide a detailed interpretation of $D(\omega \,\|\, \omega^*)$ as a one-sided kernel discrepancy, establish its connections to variational inference, and show that it is upper bounded by total variation distance under mild conditions.

Desired Properties for Ω. Because the divergence $D(\omega \,||\, \omega^*)$ is evaluated against samples drawn from a *surrogate* distribution of models Ω, its fidelity hinges on how well that ensemble of models approximates the (inaccessible) optimal distribution $p(\cdot \mid x; \omega^*)$. Three criteria follow from the definition in Eq. 3:

(i) **Support richness.** Ω covers genuinely distinct yet plausible interpretations of an input, rather than a narrow subset; otherwise the cross–similarity term in D may be artificially high, and D underestimates epistemic uncertainty when predictions of models in Ω are different from ω's predictions.

(ii) **Non-collapsing diversity.** If all members of Ω are nearly identical (e.g. noise-perturbed versions of the same model), the ensemble average would be too close to ω, hence the cross-model similarity term will be close to self-similarity and D may be small, even when the candidate predictor P_ω is *mis-specified*.

(iii) **Calibrated weighting.** Let P_Ω denote the mixing measure over models. For Eq. (3) to approach the ideal $p(y \mid x)$, each model should be weighted in proportion to its posterior credibility (e.g. uniform weights are appropriate only when validation risks are comparable).

Achieving Properties via Cross-Family Models. A practical way to satisfy these criteria is to construct the surrogate ensemble Ω from models of similar architecture and scale, likely trained on overlapping or similar pre-training datasets. Specifically, we populate Ω with 7–9B Transformer-based models that share the *same architecture class* but are trained by *different vendors*. This setup ensures (i) *support richness*, as models differ in data pipelines, initializations, and alignment protocols, which results in diverse but plausible responses for the same input, that cover the ground-truth response set. These independently trained models also exhibit (ii) *non-collapsing diversity*, as their differences arise from different design choices, rather than noise-perturbed versions of a single model. Finally, because these models achieve similar validation performance, we adopt uniform weights in P_Ω, which satisfies the *calibrated weighting* (iii) requirement. Section 4 specifies the exact models used.

Total Predictive Uncertainty. We make the standard assumption that total predictive uncertainty can be obtained by adding aleatoric and epistemic predictive uncertainties [19]. Thus, we define $U_{\text{total}}(x; \omega)$ as:

$$\begin{aligned} U_{\text{total}}(x; \omega) &= U_{\text{aleatoric}}(x; \omega) + U_{\text{epistemic}}(x; \omega) \\ &= \mathbb{E}_{\tilde{\omega} \sim P_\Omega} \mathbb{E}_{r_1^\omega \sim p(\cdot|x,\omega)} \mathbb{E}_{q_2^\omega \sim p(\cdot|x,\tilde{\omega})} \left[1 - s(r_1^\omega, q_2^{\tilde{\omega}}) \right]. \end{aligned} \tag{4}$$

3.3 Empirical Estimates of Uncertainty Metrics

For a given input prompt x, we call the model whose uncertainty is being estimated the *reference model* ω, and denote the set of models used to compute epistemic uncertainty with respect to the reference as the *auxiliary model set* Ω. Throughout the paper, we mainly focus on *Cross-family auxiliary models*: We estimate epistemic uncertainty by computing response divergence across an auxiliary set of models. To estimate uncertainty in practice, we proceed as follows:

1. Sample n responses from each model $\omega_i \in \Omega$, and denote the set of responses from ω as $R' = \{r'_1, r'_2, \ldots, r'_n\}$ and from ω_i as $R_i = \{r_1^{(i)}, r_2^{(i)}, \ldots, r_n^{(i)}\}$ where $|\Omega| = m$.
2. Approximate Aleatoric, Total, and Epistemic Uncertainty using these sampled responses.

Empirical Uncertainty Metrics

$$AU = U_{\text{aleatoric}} = 1 - \big[\sum_{k=1}^{n}\sum_{j=1}^{n} s(r'_k, r'_j)\big]/n^2$$

$$TU = U_{\text{total}} = 1 - \frac{1}{m}\sum_{i=1}^{m}\big[\sum_{k=1}^{n}\sum_{j=1}^{n} s(r'_k, r_j^{(i)})\big]/n^2$$

$$EU = U_{\text{epistemic}} = U_{\text{total}} - U_{\text{aleatoric}}$$

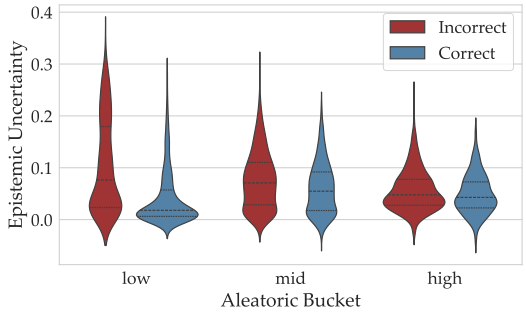
Note that we assign uniform weights to different models in the auxiliary set, as we choose to use models of similar capabilities, and our estimate of AU is similar to the one from Lin et al. [35]. Also, observe that our evaluation shows that we can keep the overall number of sampled responses at the magnitude used by self-consistency based methods while improving over those. More concretely, we choose $n = \frac{n'}{m}$, when comparing to AU as a standalone metric, where n' is the number of samples used for the latter.

4 Experimental Setup

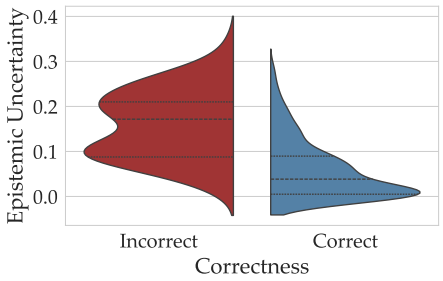
Models. In the main experiments, we primarily focus on five instruction-tuned language models with approximately 7–9B parameters: `Gemma-2-9B-It` [55], `Granite-3.0-8B-Instruct` [16], `Llama-3.1-8B-Instruct` [17], `Mistral-7B-Instruct-v0.3` [22], `Qwen2.5-7B-Instruct` [62]. We compute the uncertainty measures from Section 3.3 and consider the models mentioned above as the set of auxiliary models. In Appendix A.5, we also consider larger reference models. Unless otherwise noted, we compute TU by sampling 2 responses from each of the 5 models and similarly compute AU using 10 samples to keep the sampling budget the same across the two metrics.

Datasets. Our experiments cover a broad range of long-form generation tasks spanning question answering (QA), math reasoning, translation, and summarization. For QA, we include `AmbigQA` [40] (open-domain QA with both ambiguous and unambiguous questions), `NQ-open` [29] (closed-book QA derived from real user queries), `HotpotQA` [63] (multi-hop QA requiring reasoning over multiple supporting documents), `CoQA` [47] (conversational QA with multiple turns), `QASPER` [10] (fact-based QA over long scientific papers), `TriviaQA` [24] (QA based on trivia-style questions), and `TruthfulQA` [33] (QA to evaluate common misconceptions in models). For math reasoning, we use `GSM8K` [8] with chain-of-thought prompting. For language generation, we evaluate on the German-to-English translation dataset `WMT16-de-en` [5] and the summarization benchmark `XSum` [42]. We additionally include `SimpleQA` [59], a factuality QA benchmark, with model responses generated by `GPT-4o` [20] and `Claude 3.7 Sonnet` [1]. Finally, we adapt tasks from the BBH multiple-choice benchmark [54] to long-form format and add those evaluations to Appendix A.7.

Evaluation. Correctness is defined per input-response pair using `Meta-Llama-3-70B-Instruct` as judge (Appendix A.9). Note that, in the context of uncertainty estimation, LM-as-a-judge correctness evaluation has recently been shown to be the most reliable among the existing methods [48]. Following prior work [35, 27, 4], we evaluate the quality of uncertainty by quantifying how well uncertainty scores separate correct from incorrect generations, using Area Under the ROC Curve (AU-



(a) We stratify examples by AU (low, mid, high; 33% each) and compare the distribution of EU for correct and incorrect generations. Incorrect responses show higher EU in the low-AU regime, but this separation weakens as AU increases.



(b) We isolate the most confident samples (lowest 5% AU) and find that incorrect generations have significantly higher EU than correct ones, which shows that that EU effectively flags confidently wrong predictions.

Figure 2: The distribution of EU conditioned on aleatoric uncertainty. We observe that EU is especially discriminative in scenarios where AU is low.

ROC). Formally, AUROC corresponds to the probability that a randomly chosen incorrect response receives a higher uncertainty score than a randomly chosen correct one.

We also evaluate effectiveness in terms of selective prediction using Risk–Coverage Curves [41], which measures how the error rate changes as uncertain responses are rejected. We further report standard summary metrics such as accuracy at 90% and 80% coverage (C@90 and C@80), and the Area Under Risk-Coverage Curve (AURC), where lower is better.

Baselines. As our primary aleatoric baseline, we use Lin et al. [35]'s implementation of Semantic Entropy (SE) [27], which has been shown to perform well in recent benchmarks and surveys [12, 57]. In our evaluation, we denote this baseline as Aleatoric or AU. We also experiment with noise-perturbed models (i.e., instead of models from different model families), similar to the approach of Liu et al. [37], see details in Appendix A.5.

5 Results

5.1 Epistemic Uncertainty Flags Confident Failures of Aleatoric Uncertainty

Language models are often applied to heterogeneous tasks, where model confidence does not always align with correctness [64]. To simulate such setting, we construct an aggregated dataset by combining all datasets mentioned in Section 4, and analyze uncertainty trends on this pooled distribution. We are particularly interested in identifying where AU is low but the model is wrong, and ask whether EU can flag these instances.

In Figure 2a, we stratify examples by AU (low, mid, high) and compare EU across correct and incorrect responses. In the low-AU regime, incorrect generations consistently exhibit higher EU than correct ones, which shows that EU is discriminative when aleatoric scores are overconfident. This separation diminishes in higher AU buckets, where both correct and incorrect responses become more uncertain.

To more directly target this failure mode, we isolate the lowest 5% of AU scores and analyze EU by correctness. (Figure 2b). EU remains significantly elevated for incorrect generations, confirming that epistemic uncertainty flags confidently wrong outputs that aleatoric scores alone miss, which supports our hypothesis of the complementary nature of scores in this particular AU region.

This result contrasts with prior work, which treats low-AU predictions as reliable and only incorporates cross-model comparisons when AU exceeds a threshold [61, 7]. Our findings reveal that this assumption overlooks a critical failure mode: confidently wrong predictions with low AU. On the other hand, our observations validate findings about models being overconfident on `HotpotQA` [44] in that incorporating EU yields large improvements on this dataset (see Figure 4 in Section 5.3).

5.2 Epistemic Uncertainty, Agreement, and Diversity

We ask *when* similarity-based EU is most informative. To this end, we focus on the correctness of responses of different models and consider two metrics: *Jaccard Agreement* (or *Redundancy*) (J), which measures the overlap between predicted correct responses of the auxiliary models, which is used to quantify how redundant or similar different predictions are; and *Oracle Coverage Gain* (also *Complementarity*) (G), the additional coverage (i.e., improvement in accuracy) obtained by an oracle that always chooses the correct model per example, over the best performing model. Exact definitions can be found in Appendix A.2.

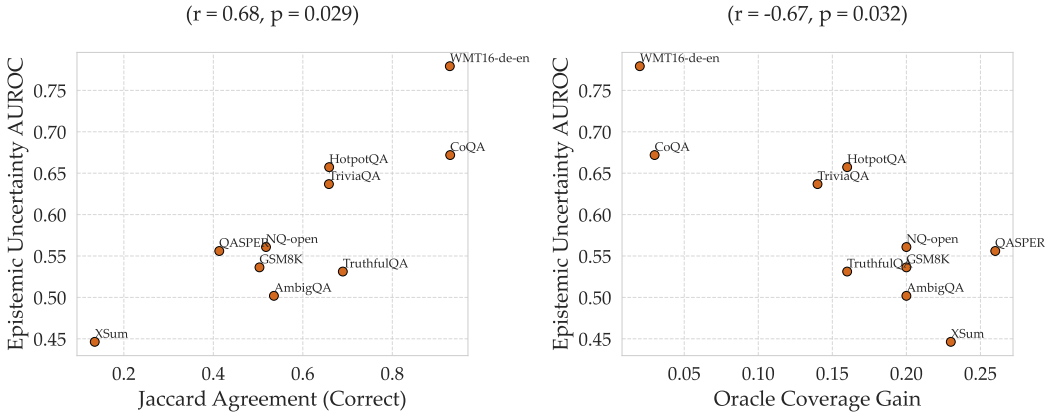


Figure 3: Epistemic–uncertainty AUROC versus dataset–level redundancy (J) and complementarity (G). Higher AUROC indicates better discrimination between correct and incorrect answers by EU.

Epistemic uncertainty does not always coincide with inter-model disagreement. Figure 3 plots EU AUROC against the dataset-level statistics J and G. We observe a positive correlation with redundancy ($r = +0.72,\ p = 0.03$) and a negative correlation with complementarity ($r = -0.72,\ p = 0.03$), which is the opposite of the naive intuition that "more disagreement $\Rightarrow$ higher epistemic utility."

The explanation lies in how EU is constructed: it grows with the *divergence of generated answers*, which arises in two distinct cases: (i) true EU on intrinsically hard questions where models do not know the answer, and (ii) the existence of many semantically different but correct responses (response noise). In complementary datasets (large G), each model specializes on different niches and consequently, EU is large even on questions that an individual model answers correctly, because models in the auxiliary set return alternative (wrong) responses. In such cases, the misalignment with correctness drives AUROC down. Conversely, in redundant datasets (high J, low G), models converge to similar responses when correct (EU low) and, what we expect to be the usual case, still diverge when collectively wrong (EU high), which gives a well-separated score and high AUROC. These observations characterize the cases in which our current EU estimator is effective: tasks with a single (or near-unique) correct answer, where models phrase that answer similarly yet generate diverse alternatives on the harder, unanswered inputs.

For example, `WMT16-de-en` and `CoQA` occupy the high-J, low-G corner of Figure 3; all models score above $> 90\%$ accuracy, so predictions are largely redundant and EU achieves its strongest discrimination. At the opposite extreme, `XSum` combines low accuracy with the largest G: models succeed on different inputs and can express many valid summaries, which inflates EU without improving ranking and thus lowering AUROC. Datasets such as `HotpotQA` and `TriviaQA` sit mid-range on both axes and have enough redundancy to suppress noise, but have sufficient diversity to expose disagreement and consequently produce the large TU gains in Figure 4.

5.3 Total Uncertainty Improves Correctness Calibration

Figure 4 reports the AUROC between negative uncertainty and correctness across datasets, and averaged over five 7–9B instruction-tuned models mentioned in Section 4. TU consistently improves over AU on all benchmarks on average. The largest gains occur on `HotpotQA` (+0.15), `CoQA` (+0.14), and `WMT16-de-en` (+0.13), where models either disagree on complex multi-hop reasoning

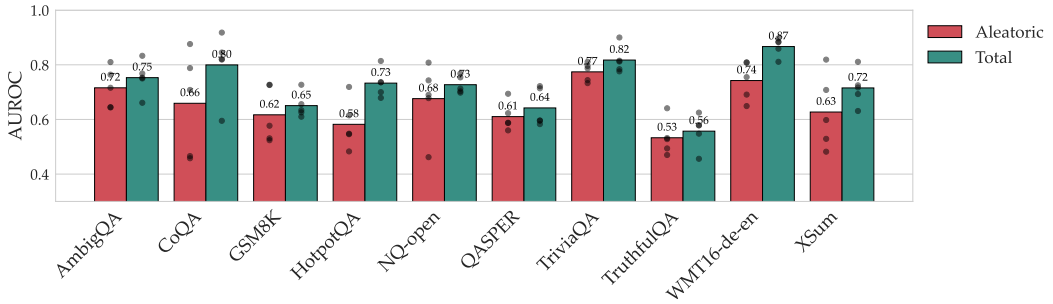


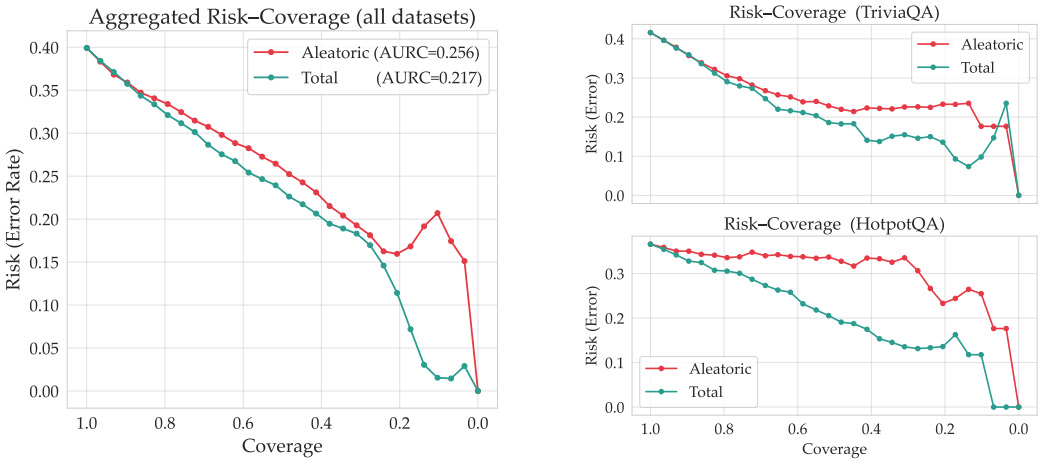
Figure 4: AUROC of aleatoric (red) and total (green) uncertainty across datasets. Bars and the text on top show the mean AUROC across five models; dots correspond to individual models. TU consistently improves discrimination between correct and incorrect outputs, with more than 0.1 improvement in AUROC in `HotpotQA`, `CoQA`, `WMT16-de-en`.

(`HotpotQA`) or achieve high overall accuracy (`CoQA`, `WMT16-de-en`), which allows EU to capture remaining errors.

Moderate improvements are observed on `TriviaQA`, and `NQ-open`, which exhibit a balance of response *redundancy* and *complementarity*. In contrast, gains are more limited on `TruthfulQA`, `GSM8K` (with chain-of-thought), and `QASPER`, where the presence of multiple valid or stylistically diverse answers weakens the alignment between TU and correctness. These results align with the patterns described in Section 5.2: TU and EU are most effective when correct answers are uniquely phrased and shared across models, while incorrect predictions remain diverse.

We also find that TU estimates consistently improve AUROC as compared to AU in `GPT-4o` (0.70 vs. 0.59), and `Claude 3.7 Sonnet` (0.58 vs. 0.53) on `SimpleQA` as shown in Figure 1. Figure 9 in Appendix A.4 shows the ROC curves on the combination of all datasets, where the relative ranking of data points across the whole dataset determines performance, and Table 2 reports AUROC per model-dataset pair. We show that the improvement over AU is maintained in individual datasets, and in the combination of all datasets.

Comparison to Other Baselines. Note that the epistemic score proposed by [2], which is repeatedly querying one model, similarly improves upon AU in such a combined dataset setting, but does not yield clear improvements over individual datasets. We show results on noise-perturbed auxiliary model sets, similar to the approach of [37], in Appendix A.5.



(a) Aggregated across all tasks. Total uncertainty yields lower error at all coverage levels.

(b) TriviaQA and HotpotQA. On both datasets, TU consistently outperforms AU.

Figure 5: Risk–coverage analysis for total (TU) versus aleatoric uncertainty (AU). TU consistently improves selective prediction across datasets and in aggregate.

Ablations. We further ablate the size of the reference model in Appendix A.5, and show that even in scenarios where the reference model is larger (and has higher accuracy) than models in the auxiliary set, TU still achieves higher AUROC than AU. Furthermore, we show that AUROC improves with a larger number of models in the auxiliary set, and sampled responses in Appendix A.6.

5.4 Total Uncertainty Improves Selective Abstention

To evaluate whether uncertainty effectively distinguishes reliable responses from potential errors, we consider selective prediction, where models are allowed to abstain from answering when uncertain.

Risk-Coverage Tradeoff. Figure 5a shows the Risk-coverage curve for aleatoric and total uncertainty, aggregated across all models mentioned in Section 4. Across all coverage levels and datasets, total uncertainty achieves the lowest risk, with a single exception. This suggests that total uncertainty more effectively identifies unreliable predictions in comparison to AU.

Selective Accuracy and AURC. To quantify this effect more precisely, Table 1 reports selective accuracy at fixed coverage levels (C@90, C@80) and AURC (area under the risk–coverage curve) across benchmarks and averaged over different models. In nearly all cases, total uncertainty achieves higher selective accuracy and in all cases lower AURC compared to aleatoric and EU alone. For example, on `HotpotQA` and `XSum`, total uncertainty improves C@90 by over 1.5 points and reduces area under the risk–coverage curve (AURC ↓) by over 20%. These results confirm that TU yields better abstention behavior than AU or EU alone.

Table 1: Selective question answering performance of different uncertainty types.

	C@90 (%)			**C@80** (%)			**AURC** ↓		
Dataset	Aleatoric	Epistemic	Total	Aleatoric	Epistemic	Total	Aleatoric	Epistemic	Total
`AmbigQA`	56.0	52.4	**56.2**	59.5	53.2	**60.2**	0.325	0.456	**0.278**
`CoQA`	96.0	**96.9**	96.0	96.5	97.5	**97.8**	0.026	0.016	**0.011**
`GSM8K`	54.0	**54.7**	53.3	**54.8**	54.0	54.2	0.391	0.441	**0.335**
`HotpotQA`	64.9	66.2	**66.9**	66.2	67.2	**69.2**	0.304	0.257	**0.206**
`NQ-open`	**53.8**	51.3	53.1	57.2	52.8	**57.5**	0.388	0.484	**0.323**
`QASPER`	38.0	36.9	**39.1**	39.5	37.5	**40.8**	0.533	0.602	**0.503**
`TriviaQA`	**64.0**	60.4	**64.0**	69.0	61.8	**70.2**	0.254	0.343	**0.208**
`TruthfulQA`	**74.4**	73.8	74.2	**75.0**	**75.0**	73.0	0.220	0.251	**0.195**
`WMT16-de-en`	96.2	**96.4**	96.0	97.2	96.2	**98.5**	0.028	0.027	**0.010**
`XSum`	24.4	24.9	**25.6**	26.5	22.0	**27.3**	0.681	0.759	**0.609**

6 Conclusions

We propose that aleatoric and epistemic uncertainty capture complementary failure modes of language models: self-consistency methods reveal data ambiguity, while semantic disagreement across models uncovers uncertainty arising from model limitations. We operationalize this view by estimating TU as the combination of intra-model entropy and inter-model semantic divergence, using only black-box access to model outputs. We show that this combination effectively outperforms self-consistency-based methods across a wide range of models and datasets in terms of different metrics. While this approach requires access to multiple comparable models, it reveals the limits of single-model uncertainty scores and offers a practical path toward more comprehensive uncertainty estimation.

Limitations. Our method relies on response-level semantic similarity, which may underperform in tasks with many semantically distinct but correct answers, e.g., open-ended generation or QA tasks where there are multiple distinct correct answers. In such cases, disagreement does not necessarily reflect uncertainty. Additionally, we focus on a specific form of aleatoric uncertainty based on semantic entropy; how to best combine epistemic uncertainty with other AU and EU estimators (e.g., token-level or logit-based methods) is left for future work. Moreover, the performance of TU depends on the model ensemble: If all surrogate models share similar pre-training data or architectural biases, cross-model disagreement can underestimate true epistemic uncertainty. We examine this homogeneous-failure scenario in detail in Section 5.2. Finally, our evaluation hinges on a correctness judge; improvements in judge reliability will propagate to more precise AUROC and selective-risk estimates.

References

[1] Claude 3.7 sonnet system card. URL https://api.semanticscholar.org/CorpusID:276612236.

[2] Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your llm: Iterative prompting for estimating epistemic uncertainty. *Advances in Neural Information Processing Systems*, 37:58077–58117, 2024.

[3] Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. Semantically diverse language generation for uncertainty estimation in language models. *arXiv preprint arXiv:2406.04306*, 2024.

[4] Neil Band, Tim GJ Rudner, Qixuan Feng, Angelos Filos, Zachary Nado, Michael W Dusenberry, Ghassen Jerfel, Dustin Tran, and Yarin Gal. Benchmarking bayesian deep learning on diabetic retinopathy detection tasks. *arXiv preprint arXiv:2211.12717*, 2022.

[5] Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. Findings of the 2016 conference on machine translation (wmt16). In *First conference on machine translation*, pages 131–198. Association for Computational Linguistics, 2016.

[6] Tapabrata Chakraborti, Christopher RS Banerji, Ariane Marandon, Vicky Hellon, Robin Mitra, Brieuc Lehmann, Leandra Bräuninger, Sarah McGough, Cagatay Turkay, Alejandro F Frangi, et al. Personalized uncertainty quantification in artificial intelligence. *Nature Machine Intelligence*, 7(4):522–530, 2025.

[7] Jianhao Chen, Zishuo Xun, Bocheng Zhou, Han Qi, Qiaosheng Zhang, Yang Chen, Wei Hu, Yuzhong Qu, Wanli Ouyang, and Shuyue Hu. Do we truly need so many samples? multi-llm repeated sampling efficiently scale test-time compute. *arXiv preprint arXiv:2504.00762*, 2025.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[9] Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. The rising costs of training frontier ai models. *arXiv preprint arXiv:2405.21015*, 2024.

[10] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. *arXiv preprint arXiv:2105.03011*, 2021.

[11] Prasenjit Dey, Srujana Merugu, and Sivaramakrishnan Kaveri. Uncertainty-aware fusion: An ensemble framework for mitigating hallucinations in large language models. *arXiv preprint arXiv:2503.05757*, 2025.

[12] Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. Lm-polygraph: Uncertainty estimation for language models. *arXiv preprint arXiv:2311.07383*, 2023.

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[14] Shiqi Gao, Tianxiang Gong, Zijie Lin, Runhua Xu, Haoyi Zhou, and Jianxin Li. Flue: Streamlined uncertainty estimation for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16745–16753, 2025.

[15] Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. Spuq: Perturbation-based uncertainty quantification for large language models. *arXiv preprint arXiv:2403.02509*, 2024.

[16] IBM Granite Team. Granite 3.0 language models, October 2024. URL https://github.com/ibm-granite/granite-3.0-language-models/.

[17] 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.

[18] HW Chung L Hou, S Longpre, B Zoph, Y Tay, W Fedus, Y Li, X Wang, M Dehghani, S Brahma, and A Webson. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

[19] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

[20] 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.

[21] Mykyta Ielanskyi, Kajetan Schweighofer, Lukas Aichberger, and Sepp Hochreiter. Addressing pitfalls in the evaluation of uncertainty estimation methods for natural language generation. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*.

[22] Albert Q Jiang, A Sablayrolles, A Mensch, C Bamford, D Singh Chaplot, Ddl Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. arxiv. *arXiv preprint arXiv:2310.06825*, 10, 2023.

[23] Daniel D Johnson, Daniel Tarlow, David Duvenaud, and Chris J Maddison. Experts don't cheat: learning what you don't know by predicting pairs. *arXiv preprint arXiv:2402.08733*, 2024.

[24] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[25] Andreas Kirsch. (implicit) ensembles of ensembles: Epistemic uncertainty collapse in large models. *arXiv preprint arXiv:2409.02628*, 2024.

[26] Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

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

[28] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023.

[29] 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.

[30] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[31] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[32] Shalev Lifshitz, Sheila A McIlraith, and Yilun Du. Multi-agent verification: Scaling test-time compute with multiple verifiers. *arXiv preprint arXiv:2502.20379*, 2025.

[33] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

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

[35] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*, 2023.

[36] Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. Uncertainty estimation and quantification for llms: A simple supervised approach. *arXiv preprint arXiv:2404.15993*, 2024.

[37] Litian Liu, Reza Pourreza, Sunny Panchal, Apratim Bhattacharyya, Yao Qin, and Roland Memisevic. Enhancing hallucination detection through noise injection. *arXiv preprint arXiv:2502.03799*, 2025.

[38] Jinliang Lu, Ziliang Pang, Min Xiao, Yaochen Zhu, Rui Xia, and Jiajun Zhang. Merge, ensemble, and cooperate! a survey on collaborative strategies in the era of large language models. *arXiv preprint arXiv:2407.06089*, 2024.

[39] Huan Ma, Jingdong Chen, Guangyu Wang, and Changqing Zhang. Estimating llm uncertainty with logits. *arXiv preprint arXiv:2502.00290*, 2025.

[40] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*, 2020.

[41] Malik Sajjad Ahmed Nadeem, Jean-Daniel Zucker, and Blaise Hanczar. Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. In *Machine Learning in Systems Biology*, pages 65–81. PMLR, 2009.

[42] Shashi Narayan, Shay B Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*, 2018.

[43] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*, 2021.

[44] Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. *arXiv preprint arXiv:2402.11457*, 2024.

[45] 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.

[46] Lorenzo Pacchiardi, Alex J Chan, Sören Mindermann, Ilan Moscovitz, Alexa Y Pan, Yarin Gal, Owain Evans, and Jan Brauner. How to catch an ai liar: Lie detection in black-box llms by asking unrelated questions. *arXiv preprint arXiv:2309.15840*, 2023.

[47] 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.

[48] Andrea Santilli, Adam Golinski, Michael Kirchhof, Federico Danieli, Arno Blaas, Miao Xiong, Luca Zappella, and Sinead Williamson. Revisiting uncertainty quantification evaluation in language models: Spurious interactions with response length bias results. *arXiv preprint arXiv:2504.13677*, 2025.

[49] Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. Introducing an improved information-theoretic measure of predictive uncertainty. *arXiv preprint arXiv:2311.08309*, 2023.

[50] Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

[51] Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *arXiv preprint arXiv:2412.05563*, 2024.

[52] Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. Trust me, i'm wrong: High-certainty hallucinations in llms. *arXiv preprint arXiv:2502.12964*, 2025.

[53] Lintang Sutawika, Leo Gao, Hailey Schoelkopf, Stella Biderman, Jonathan Tow, Baber Abbasi, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas Muennighoff, Thomas Wang, sdtblck, nopperl, gakada, tttyuntian, researcher2, Chris, Julen Etxaniz, Zdeněk Kasner, Khalid, Jeffrey Hsu, AndyZwei, Pawan Sasanka Ammanamanchi, Dirk Groeneveld, Ethan Smith, and Eric Tang. Eleutherai/lm-evaluation-harness: Major refactor, December 2023. URL https://doi.org/10.5281/zenodo.10256836.

[54] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[55] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[56] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[57] Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, et al. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248, 2025.

[58] Xi Wang, Laurence Aitchison, and Maja Rudolph. Lora ensembles for large language model fine-tuning. *arXiv preprint arXiv:2310.00035*, 2023.

[59] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

[60] Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. A survey of uncertainty estimation methods on large language models. *arXiv preprint arXiv:2503.00172*, 2025.

[61] Yihao Xue, Kristjan Greenewald, Youssef Mroueh, and Baharan Mirzasoleiman. Verify when uncertain: Beyond self-consistency in black box hallucination detection. *arXiv preprint arXiv:2502.15845*, 2025.

[62] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[63] 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. *arXiv preprint arXiv:1809.09600*, 2018.

[64] Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. Larger and more instructable language models become less reliable. *Nature*, 634(8032):61–68, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Methods, and Results section include results for each claim.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the limitations paragraph in conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: In Section 3.2 and Appendix A.1.

   Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details provided in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use public datasets, and will submit our code with the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Results we include in the main paper are averaged across datasets or models, but we include error bars in Appendix A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Appendix A.3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Discussed briefly in conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not propose new models or datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Cited throughout the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: Used only for editing and plotting.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Appendix

A.1 Theoretical Interpretations of Epistemic Uncertainty

Kernel and variational interpretation of $D(\omega \,||\, \omega^*)$. Assume the similarity function $s(\cdot,\cdot)$ is a symmetric positive definite kernel k. Denote the predictive distributions by $P_\Omega := p(\cdot \mid x;\omega)$ and P_{ω^*}. Their kernel mean embeddings in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ are

$$\mu_\omega \;=\; \mathbb{E}_{r\sim P_\Omega}\big[k(r,\cdot)\big], \qquad \mu_{\omega^*} \;=\; \mathbb{E}_{q\sim P_{\omega^*}}\big[k(q,\cdot)\big].$$

Using the reproducing property $\langle k(r,\cdot), k(r',\cdot)\rangle_{\mathcal{H}_k} = k(r,r')$, the divergence in Eq. (2) can be rewritten exactly as:

$$D(\omega \,||\, \omega^*) = \langle \mu_\omega, \mu_\omega\rangle_{\mathcal{H}_k} \;-\; \langle \mu_\omega, \mu_{\omega^*}\rangle_{\mathcal{H}_k}. \tag{5}$$

Equation (5) is the first two terms of the squared maximum mean discrepancy (MMD):

$$\mathrm{MMD}^2(P_\Omega, P_{\omega^*}) = \|\mu_\omega - \mu_{\omega^*}\|^2_{\mathcal{H}_k} = \underbrace{\|\mu_\omega\|^2_{\mathcal{H}_k} - \langle \mu_\omega, \mu_{\omega^*}\rangle_{\mathcal{H}_k}}_{D(\omega \,||\, \omega^*)} + \|\mu_{\omega^*}\|^2_{\mathcal{H}_k} - \langle \mu_\omega, \mu_{\omega^*}\rangle_{\mathcal{H}_k}.$$

Thus $D(\omega \,||\, \omega^*)$ is a *one-sided kernel discrepancy*: it measures how much the model's self-agreement exceeds its agreement with the ideal predictor, and it vanishes if and only if $\mu_\omega = \mu_{\omega^*}$ (and, for characteristic kernels, iff $P_\Omega = P_{\omega^*}$).

Variational-gap Interpretation. Write classical KL as $\mathrm{KL}(P_\Omega \,||P_{\omega^*}) = \mathrm{CE} - \mathrm{Ent}$, where $\mathrm{Ent} = \mathbb{E}_{r\sim P_\Omega}[-\log p(r \mid x;\omega)]$ and $\mathrm{CE} = \mathbb{E}_{r\sim P_\Omega}[-\log p(r \mid x)]$. Replacing $-\log$ with $-k$ yields

$$D(\omega \,||\, \omega^*) \;=\; \underbrace{\mathbb{E}_{r,r'\sim P_\Omega}[k(r,r')]}_{\text{“negative kernel-entropy”}} \;-\; \underbrace{\mathbb{E}_{r\sim P_\Omega,\, q\sim P_{\omega^*}}[k(r,q)]}_{\text{“kernel cross-entropy”}},$$

so D is the *semantic variational gap* between the model and the ideal distribution under the geometry induced by k. Minimizing D therefore projects P_Ω toward P_{ω^*} in RKHS while simultaneously penalizing model variability in P_Ω.

Lemma 1 (Bound on Kernel-Based Divergence)**.** *Let $P_\omega = p(\cdot \mid x,\omega)$ and $P_{\omega^*} = p(\cdot \mid x,\omega^*)$ be the predictive distributions of two language models. Let $k(\cdot,\cdot)$ be a symmetric, bounded, positive-definite kernel such that $0 \le k(r,r') \le 1$ for all r, r'. Define the kernel-based divergence*

$$D(\omega \,\|\, \omega^*) := \mathbb{E}_{r,r'\sim P_\omega}[k(r,r')] - \mathbb{E}_{r\sim P_\omega, q\sim P_{\omega^*}}[k(r,q)].$$

Then D is bounded above in absolute value by the total variation distance between P_ω and P_{ω^*}:

$$|D(\omega \,\|\, \omega^*)| \le \mathrm{TVD}(P_\omega, P_{\omega^*}),$$

where

$$\mathrm{TVD}(P_\omega, P_{\omega^*}) := \frac{1}{2}\int |p_\omega(z) - p_{\omega^*}(z)|dz.$$

Furthermore, if the RKHS norm of the embeddings are equal, i.e., $\|\mu_\omega\| = \|\mu_{\omega^*}\|$, then $D(\omega \,\|\, \omega^*) = 0$ implies $\mu_\omega = \mu_{\omega^*}$. If k is characteristic, this further implies $P_\omega = P_{\omega^*}$.

Proof. Define the kernel-smoothed function $f(z) := \mathbb{E}_{r\sim P_\omega}[k(r,z)]$. Then we can write

$$D(\omega \,\|\, \omega^*) = \mathbb{E}_{z\sim P_\omega}[f(z)] - \mathbb{E}_{z\sim P_{\omega^*}}[f(z)].$$

By the definition of total variation distance and the fact that $0 \le f(z) \le 1$ for all z,

$$|D(\omega \,\|\, \omega^*)| \le \sup_{\|f\|_\infty \le 1} |\mathbb{E}_{P_\omega}[f] - \mathbb{E}_{P_{\omega^*}}[f]| = \mathrm{TVD}(P_\omega, P_{\omega^*}).$$

For the second part, note that $D(\omega \,\|\, \omega^*) = \|\mu_\omega\|^2 - \langle \mu_\omega, \mu_{\omega^*}\rangle$. If $\|\mu_\omega\| = \|\mu_{\omega^*}\|$ and $D = 0$, then by Cauchy–Schwarz equality, we must have $\mu_\omega = \mu_{\omega^*}$. If the kernel is characteristic, then $\mu_\omega = \mu_{\omega^*}$ implies $P_\omega = P_{\omega^*}$. □

This result implies that $D(\omega \,\|\, \omega^*)$ provides a lower bound on distributional mismatch, and thus serves as a tractable proxy for epistemic uncertainty: it is provably small only when the model's predictive distribution aligns with that of the ensemble under the kernel geometry.

Under desired conditions mentioned in 3.2, and with a bounded characteristic kernel, $D(\omega \,||\, \omega^*) = 0$ if, and only if, the predictive distribution P_ω is close to the model set consensus as shown above. Consequently D (i) flags cases where the model's intra-model similarity is high (AU is low) while its agreement with the ensemble is low, (ii) rewards calibrated agreement by attaining its minimum only within the support of the ensemble, and (iii) remains sample-efficient, as it only requires $\mathcal{O}(n^2)$ kernel evaluations on n generated responses per model, which is cheaper than KL divergence evaluations or other approximations of EU.

A.2 Agreement and Coverage Metrics.

For every dataset we form the binary correctness matrix C_{ij} which is the correctness of response $r_i^{(j)}$ sampled from model j to input i, and compute two coarse descriptors of cross-model behaviour:

- **Jaccard redundancy** J – The mean pairwise Jaccard index of the sets of correctly answered examples:

$$J = \frac{2}{M(M-1)} \sum_{1 \le m < k \le M} \frac{|S_m \cap S_k|}{|S_m \cup S_k|}, \quad S_m = \{i : C_{im} = 1\}$$

  High J means models succeed on the same inputs.

- **Oracle-gain diversity** G – the additional coverage obtained by an oracle that chooses the correct model per example:

$$G = A_{\text{oracle}} - \max_m A_m = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left[\sum_{j=1}^{M} C_{ij} > 0\right] - \max_m \left(\frac{1}{N} \sum_{i=1}^{N} C_{im}\right)$$

  High G indicates that different models get different examples right.

A.3 Additional Experimental Setup

All experiments were conducted on two NVIDIA A100 80GB GPUs. We use the `lm-evaluation-harness` [53] codebase to generate responses from each model, and evaluate correctness using a local vLLM [30] server hosting `Meta-Llama-3-70B-Instruct` [17] as the judge model. Following prior work [35], we compute correctness using only the first sampled response from each model. All evaluation is conducted in inference-only mode; no training or fine-tuning is performed.

For each dataset, we sample 10 responses per model for the first 100 prompts. AU is computed using all 10 responses. To match the sample budget with TU, the experiments in Section 5.3 and Section 5.4 report results using only 2 responses per model for computing both TU and epistemic uncertainty. We use a temperature of 0.7 and top-p of 0.9 across all generations. For `SimpleQA`, model outputs are obtained from the OpenAI and Anthropic APIs.

Semantic similarity between responses is measured using cosine distance over `sentence-T5-xl` [18, 43] embeddings. For datasets not originally supported by `lm-eval-harness`, we follow its prompt formatting conventions and include code for these additions in the supplementary material.

A.4 Additional Results on Total Uncertainty

Figure 6 reports the AUROC of aleatoric and total uncertainty across all model–dataset pairs, and Figure 7 shows the corresponding improvements from using TU over AU. Total uncertainty consistently improves correctness discrimination in nearly all cases, with the largest per-instance gains observed on `HotpotQA`, a benchmark known for complex multi-hop reasoning.

AUROC Results Per Model and Dataset. Figure 6 and Table 2 show AUROC of TU and AU in all model-dataset combinations, and Figure 7 shows the improvement of total uncertainty over AU in AUROC. TU improves AUROC in nearly all model–dataset combinations, with the largest gains

observed in HotpotQA. Model performance substantially affects the magnitude of improvement. On datasets such as XSum, where overall model accuracy is low, TU yields large improvements for weaker models but occasionally underperforms for the strongest ones (e.g., Llama and Qwen), potentially due to disagreement with less reliable auxiliary models. A similar pattern holds on GSM8K, where gains are concentrated among lower-performing models, while others benefit less.

In contrast, WMT16-de-en and CoQA show limited gains, likely due to the high baseline accuracy ($> 90\%$) of all reference models (see Fig. 8), where AU is already well-calibrated. Notably, TU corrects miscalibrated AU for specific models, such as Mistral on CoQA, where the base AU is anomalously low. On TruthfulQA, which features open-ended questions with diverse valid answers, semantic disagreement does not reliably indicate epistemic uncertainty, which results in weaker improvements.

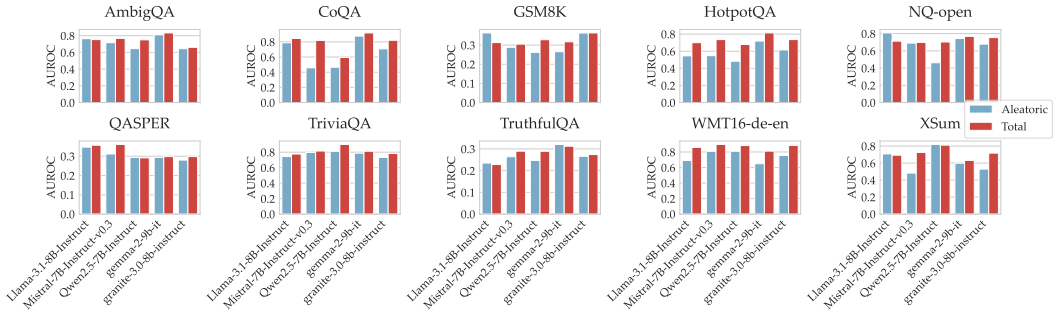


Figure 6: We show AUROC for each model separately to compare aleatoric and total uncertainty. TU consistently yields higher AUROC across models.

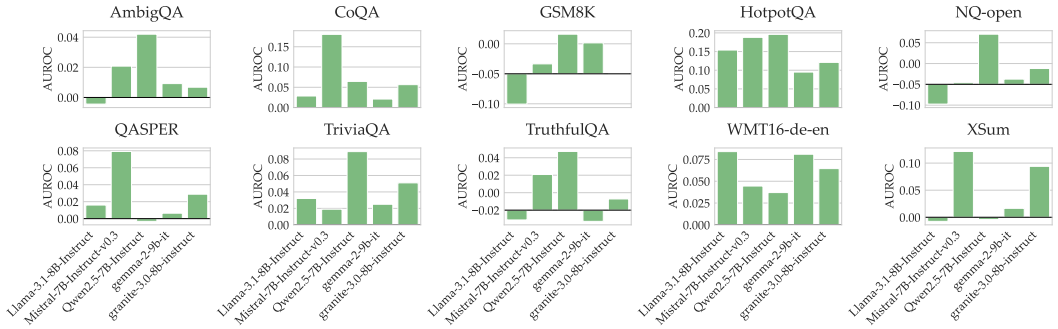


Figure 7: AUROC improvement obtained by adding EU to AU, measured as (Total − Aleatoric).

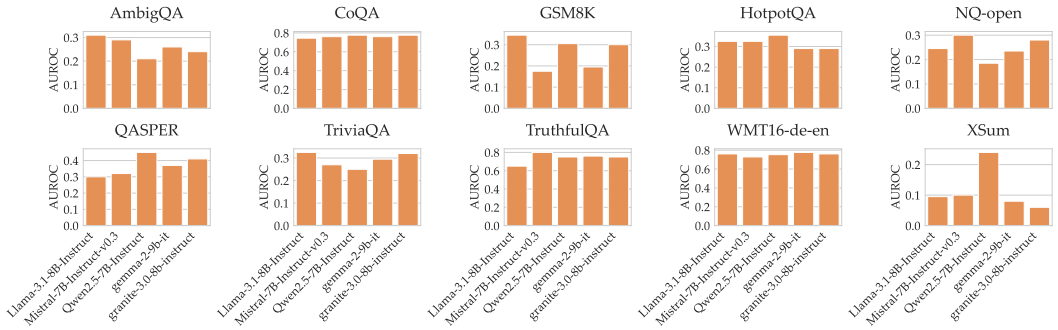


Figure 8: Accuracy per model-dataset pair.

Table 2: Uncertainty AUROC scores across models and benchmarks when different 7B/8B/9B parameter models are used as auxiliary models. Total Uncertainty is better calibrated with correctness than Aleatoric Uncertainty.

Benchmark	Model	Accuracy	Aleatoric AUROC	Epistemic AUROC	Total AUROC
AmbigQA	Llama-3.1-8B-Instruct	0.62	**0.764**	0.52	0.753
	Mistral-7B-Instruct-v0.3	0.58	0.716	0.506	**0.768**
	Qwen2.5-7B-Instruct	0.42	0.645	0.639	**0.75**
	gemma-2-9b-it	0.52	0.81	0.447	**0.833**
	granite-3.0-8b-instruct	0.48	0.644	0.488	**0.661**
CoQA	Llama-3.1-8B-Instruct	0.93	0.788	0.797	**0.845**
	Mistral-7B-Instruct-v0.3	0.95	0.458	0.80	**0.819**
	Qwen2.5-7B-Instruct	0.97	0.466	0.567	**0.595**
	gemma-2-9b-it	0.95	0.876	0.80	**0.918**
	granite-3.0-8b-instruct	0.97	0.708	**0.907**	0.821
GSM8K	Llama-3.1-8B-Instruct	0.69	**0.727**	0.346	0.626
	Mistral-7B-Instruct-v0.3	0.35	0.577	0.606	**0.61**
	Qwen2.5-7B-Instruct	0.61	0.524	0.623	**0.656**
	gemma-2-9b-it	0.39	0.531	0.601	**0.634**
	granite-3.0-8b-instruct	0.60	0.726	0.472	**0.727**
HotpotQA	Llama-3.1-8B-Instruct	0.65	0.546	0.662	**0.7**
	Mistral-7B-Instruct-v0.3	0.65	0.548	0.682	**0.736**
	Qwen2.5-7B-Instruct	0.71	0.483	**0.681**	0.679
	gemma-2-9b-it	0.58	0.719	0.608	**0.814**
	granite-3.0-8b-instruct	0.58	0.615	0.634	**0.736**
NQ-open	Llama-3.1-8B-Instruct	0.49	**0.808**	0.389	0.713
	Mistral-7B-Instruct-v0.3	0.60	0.69	0.501	**0.698**
	Qwen2.5-7B-Instruct	0.37	0.462	0.696	**0.703**
	gemma-2-9b-it	0.47	0.743	0.538	**0.768**
	granite-3.0-8b-instruct	0.56	0.678	0.541	**0.754**
QASPER	Llama-3.1-8B-Instruct	0.30	0.694	0.487	**0.714**
	Mistral-7B-Instruct-v0.3	0.32	0.623	0.657	**0.722**
	Qwen2.5-7B-Instruct	0.45	**0.587**	0.492	0.583
	gemma-2-9b-it	0.37	0.588	0.509	**0.596**
	granite-3.0-8b-instruct	0.41	0.56	0.512	**0.596**
TriviaQA	Llama-3.1-8B-Instruct	0.65	0.744	0.562	**0.776**
	Mistral-7B-Instruct-v0.3	0.54	0.796	0.552	**0.815**
	Qwen2.5-7B-Instruct	0.5	0.811	0.668	**0.9**
	gemma-2-9b-it	0.59	0.787	0.645	**0.812**
	granite-3.0-8b-instruct	0.64	0.733	0.575	**0.784**
TruthfulQA	Llama-3.1-8B-Instruct	0.65	**0.47**	0.423	0.456
	Mistral-7B-Instruct-v0.3	0.80	0.528	**0.601**	0.579
	Qwen2.5-7B-Instruct	0.75	0.494	**0.584**	0.578
	gemma-2-9b-it	0.76	**0.641**	0.541	0.625
	granite-3.0-8b-instruct	0.75	0.532	**0.556**	0.548
WMT16-de-en	Llama-3.1-8B-Instruct	0.95	0.691	0.695	**0.859**
	Mistral-7B-Instruct-v0.3	0.91	0.808	0.835	**0.897**
	Qwen2.5-7B-Instruct	0.94	0.809	0.832	**0.883**
	gemma-2-9b-it	0.97	0.649	0.663	**0.811**
	granite-3.0-8b-instruct	0.95	0.755	0.493	**0.884**
XSum	Llama-3.1-8B-Instruct	0.19	**0.708**	0.37	0.693
	Mistral-7B-Instruct-v0.3	0.20	0.482	0.651	**0.725**
	Qwen2.5-7B-Instruct	0.48	**0.819**	0.31	0.811
	gemma-2-9b-it	0.16	0.598	0.565	**0.631**
	granite-3.0-8b-instruct	0.12	0.529	0.582	**0.717**

ROC Curves. Figure 9 shows the ROC curve computed over the pooled set of all model–dataset pairs. TU achieves a higher AUROC (0.746 vs. 0.707), which shows improved seperation between correct and incorrect generations compared to AU alone. Figure 10 presents ROC curves for individual datasets. TU yields consistently better or comparable performance across all tasks, with the largest gains observed on `HotpotQA`, `WMT16-de-en`, and `CoQA`. These improvements align with our earlier findings that TU is most effective on tasks where models are accurate but occasionally confidently wrong.

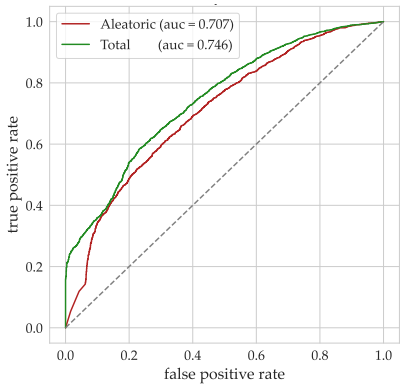


Figure 9: ROC curves between aleatoric and total uncertainty aggregated across all models and datasets. Total uncertainty achieves higher AUROC, indicating better discrimination between correct and incorrect generations.

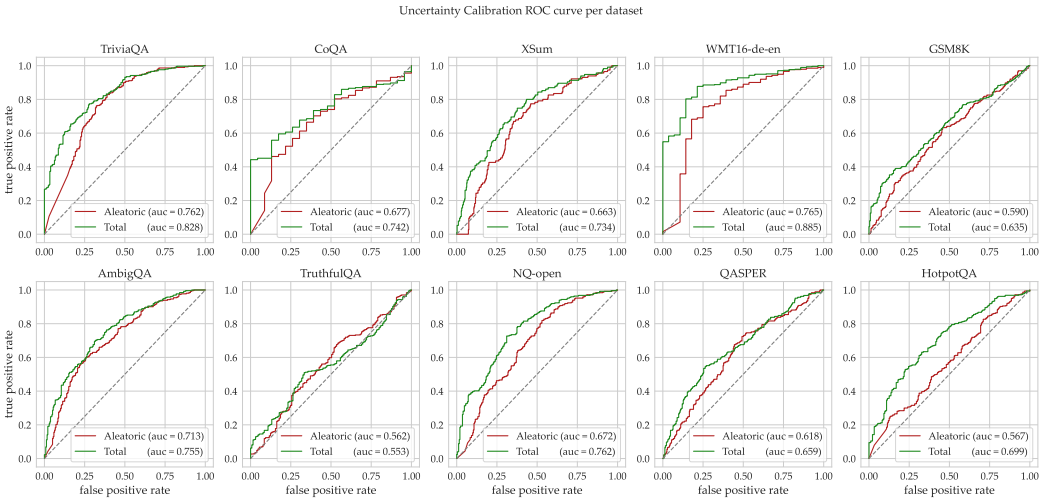


Figure 10: ROC curves comparing aleatoric and total uncertainty across individual datasets. TU achieves higher AUROC across most tasks, particularly on `HotpotQA`, `WMT16-de-en`, and `CoQA`, where models exhibit confident failures.

Additional Baseline. The results in Table 3 illustrate the variability in token-level AUROC scores across different uncertainty formulations from Ma et al. [39]. While mean token entropy performs well on tasks like `XSum`, where stylistic variation is high, the probability disparity score performs poorly on structured, factual tasks, which suggests that it may conflate EU with token frequency effects. This further motivates the need for sequence-level uncertainty measures like ours, which better align with correctness across diverse generation tasks.

Dataset	Mean Token Entropy (AUROC)	Probability Disparity (AUROC)
CoQA	0.671	0.723
NQ-open	0.653	0.575
TriviaQA	0.736	0.728
WMT-de-en	0.877	0.518
XSum	0.934	0.524

Table 3: Comparison of token-level uncertainty scores from Ma et al. [39]. Mean token entropy and probability disparity are derived from token logits without sampling. While both achieve strong performance on some datasets (e.g., `XSum`), probability disparity underperforms on factual tasks such as `CoQA`.

A.5 The Effect of the Auxiliary Model Set on Total Uncertainty

Ablating the Reference Model Size. We test how the quality of total uncertainty estimates depends on the capability of the *reference model*, whose uncertainty we aim to estimate. We fix the auxiliary model set to a pool of four 7-9B models mentioned in 4 that are not from the same family as the reference model, and vary the reference model's architecture and size. Figure 11 reports results on `TriviaQA`, using two model families (`Gemma3` [56] and `Qwen2.5` [62]) of various sizes. As the size of the reference model increases, both aleatoric and total uncertainty AUROC scores tend to decrease, but total uncertainty has consistently higher AUROC across different model sizes. This holds even when the reference model is substantially stronger than any model in the auxiliary set (e.g., `Qwen2.5-32B` vs. 7-9B peers).

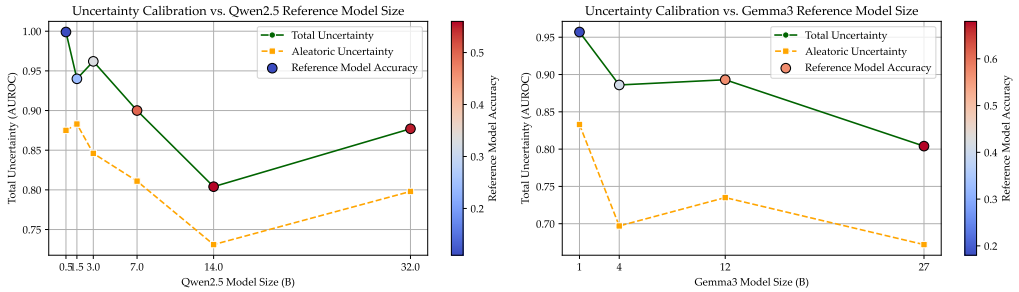


Figure 11: We vary the size of the **reference model** while holding the auxiliary models fixed. TU achieves higher AUROC in comparison to AU acorss different model sizes on `TriviaQA`.

Ablating Auxiliary Model Size. We pick a reference model (mistral), and let the auxiliary set be a single model ranging from 0.5B to 32B Qwen2.3 Model and x to yB parameter Gemma 3 model. We find that larger model sizes contribute to

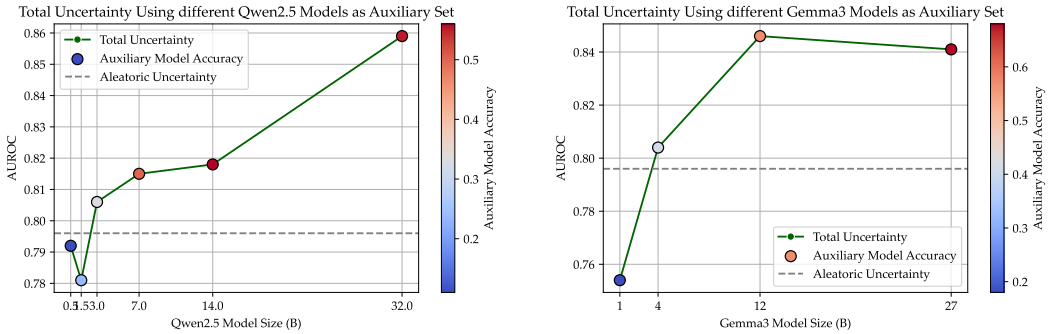


Figure 12: We keep the reference model fixed as `mistral`, and vary the size of the **auxiliary model**. TU achieves higher AUROC in comparison to AU with larger and more capable the auxiliary model on `TriviaQA`.

Noise-perturbed Auxiliary Model Set. We consider noise-perturbed variants of the reference model itself as auxiliary model set, similar to Liu et al. [37]. Specifically, we apply a perturbation strategy in which we preserve the top-k singular vectors of each linear weight matrix and inject Gaussian noise into the remaining lower-rank subspace. This preserves dominant components of the model while allowing for controlled noise in its response distribution. Figure 13 in the appendix shows that the we sometimes obtain improvement in TU-AU, but it is overall lower than for the more diverse auxiliary model set from Figure 4.

A.6 AUROC Ablations

Number of Auxiliary Models. We study how the size of the auxiliary model set affects the quality of total uncertainty estimates. For each reference model, we compute total uncertainty using $n \in \{2, 3, 4, 5\}$ models, where one model is fixed (the reference model) and the remaining $n - 1$ are sampled from other model families. All methods use a fixed number of samples per model.

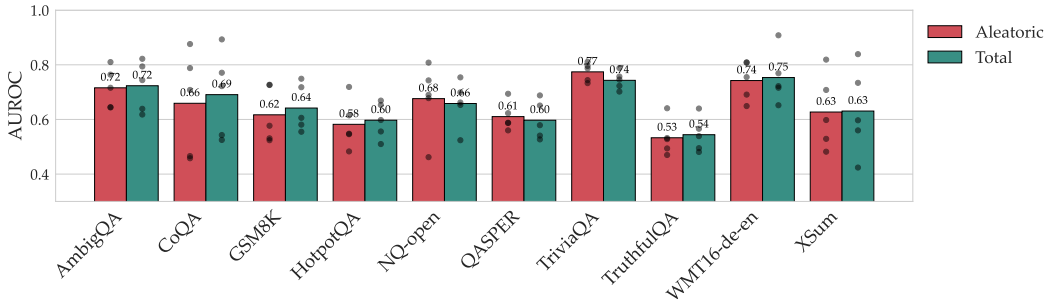


Figure 13: Uncertainty calibration for experiments where auxiliary model set for each model is consisted of multiple noise-perturbed models

Figure 14 shows that total uncertainty improves monotonically as the number of auxiliary models increases. This holds across almost all tasks, with the largest gains typically occurring between $n = 2$ and $n = 3$. In addition, we observe that variance across runs decreases as more models are added, which suggests a more calibrated uncertainty score can be achieved from increasing the number of model in the auxiliary set. However, in all datasets, our multi-sample total uncertainty measure outperforms aleatoric uncertainty in AUROC, even when only one auxiliary model is used.

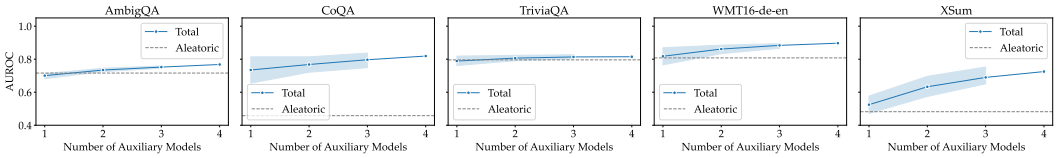


Figure 14: We plot AUROC as a function of the number of auxiliary models used to compute total uncertainty for `Mistral-7B-Instruct-v0.3`. Total uncertainty improves with more models, and variance decreases.

Number of Samples for Uncertainty Estimation. We next investigate how the number of response samples per model affects the performance of uncertainty estimates. For each model in the auxiliary set, we vary the number of generations used in total uncertainty computation from 5 to 50, and compare against two baselines for aleatoric uncertainty: one computed using 5 samples and another using 10 samples, matching the regimes used in our main experiments.

As shown in Figure 15, AUROC for total uncertainty usually slightly increases with more samples, with diminishing returns beyond 30 samples in most tasks. Notably, TU consistently outperforms AU baselines across all datasets. These findings also reinforce the practicality of TU even under constrained budgets, as improvements are apparent with as few as 10 samples ($n = 5$ on the x-axis).

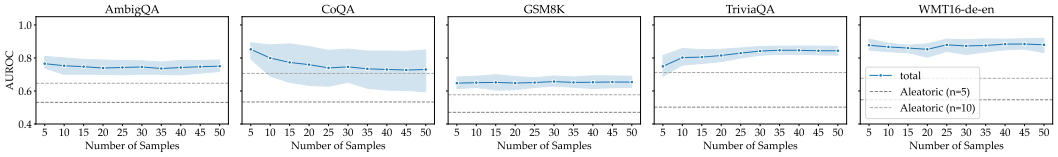


Figure 15: AUROC of total uncertainty as a function of the number of samples per model. Even with a small number of samples, TU outperforms aleatoric baselines (5-sample and 10-sample variants). Gains saturate around 30–40 samples.

A.7 Additional Results on Multiple-choice QA tasks

To evaluate whether our findings extend beyond open-ended generation, we adapt a subset of tasks from the Big-Bench Hard (BBH) [54] benchmark into a long-form QA format with chain-of-thought answering. Specifically, we consider `Boolean Expressions`, `Disambiguation QA`, and `Word Sorting`, and prompt models to justify their answers rather than selecting from multiple choices directly. We then evaluate uncertainty scores over the full responses using the same semantic similarity pipeline as in our main experiments.

Table 4 reports AUROC scores for both AU and TU across models on these tasks. We observe that TU improves over AU in most cases, with the largest gains appearing when base model performance is low (e.g., Qwen2.5-7B on `Disambiguation QA` and `Boolean Expressions`). These results demonstrate that TU remains effective in identifying incorrect generations even when the task is originally framed as multiple-choice, provided responses are elicited in free-form.

Table 4: Uncertainty AUROC scores across models and benchmarks when different 7B/8B/9B parameter models are used as auxiliary models. Total Uncertainty is better calibrated with correctness than Aleatoric Uncertainty.

Benchmark	Model	Accuracy	Aleatoric AUROC	Total AUROC
BBH Fewshot Boolean Expressions	`Llama-3.1-8B-Instruct`	**0.88**	**0.662**	0.658
	`Mistral-7B-Instruct-v0.3`	**0.84**	**0.746**	0.735
	`Qwen2.5-7B-Instruct`	**0.53**	0.744	**0.909**
	`gemma-2-9b-it`	**0.86**	0.593	**0.725**
	`granite-3.0-8b-instruct`	**0.9**	**0.659**	0.658
BBH Fewshot Disambiguation QA	`Llama-3.1-8B-Instruct`	**0.59**	0.544	**0.594**
	`Mistral-7B-Instruct-v0.3`	**0.64**	0.525	**0.656**
	`Qwen2.5-7B-Instruct`	**0.44**	0.561	**0.81**
	`gemma-2-9b-it`	**0.69**	0.61	**0.65**
	`granite-3.0-8b-instruct`	**0.62**	0.486	**0.562**
BBH Fewshot Word Sorting	`Llama-3.1-8B-Instruct`	**0.69**	0.476	**0.512**
	`Mistral-7B-Instruct-v0.3`	**0.77**	**0.529**	0.429
	`Qwen2.5-7B-Instruct`	**0.44**	0.587	**0.645**
	`gemma-2-9b-it`	**0.96**	0.475	**0.576**
	`granite-3.0-8b-instruct`	**0.58**	**0.578**	0.485

A.8 Epistemic Uncertainty Analysis

Figure 16 disaggregates the trend shown in Figure 2a by model. For all five reference models, incorrect generations in the low-AU regime show consistently higher EU than correct ones, which reaffirms that EU captures confident failures missed by self-consistency. This separation weakens in mid- and high-AU buckets, where both correct and incorrect outputs tend to be more uncertain. The consistency of this pattern across different models highlights the effectiveness of EU in identifying unreliable predictions when AU alone is low.

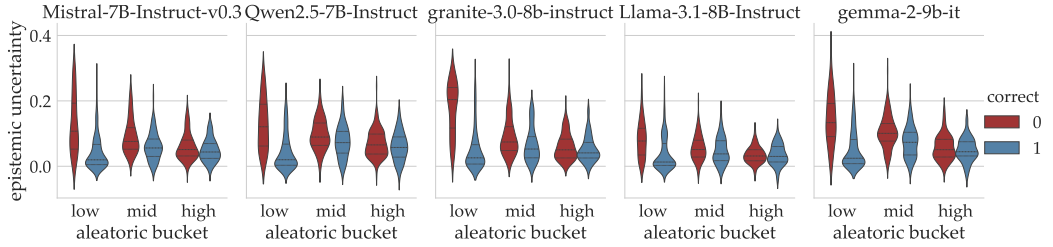


Figure 16: Distribution of EU across different levels of AU and correctness. Across all models, we find that incorrect responses in the low-aleatoric regime are assigned higher EU than correct ones on average.

A.9 Computing Correctness Using an LLM Judge

We compute correctness scores using `Meta-Llama-3-70B-Instruct` deployed via a local `vLLM` server. Each model prediction is evaluated independently against the gold answers using a structured prompt that includes five few-shot examples, held fixed across all evaluations. The prompt instructs the judge to assign a correctness score from the discrete set `{0.0, 0.1, ..., 1.0}` based on

the alignment between the predicted and gold answers, while explicitly ignoring the model's own knowledge.

The judge receives as input: (1) the user-defined task or question, (2) a list of gold answers, and (3) the model-generated answer. It is instructed to output a JSON object containing a numerical score and a justification. The request is submitted using deterministic decoding (`temperature`= 0, `max_tokens` = 20), and we employ up to three retries with truncated context in case of failures due to prompt length.

Correctness is evaluated using the first response generated by each model. The gold answers are passed verbatim, and no normalization is applied to either predictions or references. By default, a prediction is considered correct if its score exceeds 0.5. For tasks where differences should be penalized (e.g., summarization or translation), we increase the threshold to 0.9 (specifically, for `XSum` and `WMT16-de-en`). These thresholds are applied during AUROC and selective prediction evaluations.

Prompt Format. The full judge prompt used in all evaluations is shown below (example QA pairs are fixed across all examples):

```
I want you to act as a judge for how well a model did answering a user-defined
task.
You will be provided with a user-defined task that was given to the model, its
golden answer(s), and the model's answer.  The context of the task may not be
given here.  Your task is to judge how correct the model's answer is based on
the golden answer(s), without seeing the context of the task, and then give a
correctness score.  The correctness score should be one of the below numbers:
0.0 (totally wrong), 0.1, 0.2, ..., 1.0 (totally right).  You should also add a
brief justification regarding how the model's answer conforms to or contradicts
the golden answer(s).  Your response must follow the format:
{
"correctness_score":  your_score,
"justification":  your_justification
}
Note that each one of the golden answers is considered correct.  Thus if the
model's answer matches any one of the golden answers, it should be considered
correct.
--
Example 1:
User-defined task -- Sandy bought 1 million Safe Moon tokens.  She has 4
siblings.  She wants to keep half of them to herself and divide the remaining
tokens among her siblings.  After splitting it up, how many more tokens will she
have than any of her siblings?
Golden Answer(s) -- <answer 1> 375000
Model's Answer -- Sandy will have more tokens than any sibling by 3/8 million.
Model Output:
{
"correctness_score":  1.0,
"justification":  "The model's answer of 3/8 million equals 375,000, which
matches the gold answer exactly."
}
--
...  (3 more examples)
--
Target Example:
User-defined task -- [QUESTION]
Golden Answer(s) -- <answer 1> [...]; <answer 2> [...]
Model's Answer -- [MODEL RESPONSE]
Model Output:
{
"correctness_score":  ?,
"justification":  ?
}
```