# Why Don't You Know? Evaluating the Impact of Uncertainty Sources on Uncertainty Quantification in LLMs

Anonymous ACL submission

## Abstract

As Large Language Models (LLMs) are increasingly deployed in real-world applications, reliable uncertainty quantification (UQ) becomes critical for safe and effective use. Most existing UQ approaches for language models aim to produce a single confidence score – for example, estimating the probability that a model's answer is correct. However, uncertainty in natural language tasks arises from multiple distinct sources, including model knowledge gaps, output variability, and input ambiguity, which have different implications for system behavior and user interaction. In this work, we study how the source of uncertainty impacts the behavior and effectiveness of existing UQ methods. To enable controlled analysis, we construct a human-validated dataset that introduces distinct uncertainty sources in a controlled way, enabling the first systematic evaluation of how uncertainty source affects UQ method behavior. Our experiments reveal that method families fail in predictable, source-specific ways, and that no single method reliably handles all sources. These findings highlight the need for uncertainty-aware methods that explicitly account for the source of uncertainty in large language models. Our data and code are available at http://anonymous.for.review.

## 1 Introduction

Large Language Models (LLMs) are increasingly deployed across a variety of domains, including high stakes ones, like medicine, law and education. Despite their impressive capabilities, they remain prone to producing factually incorrect generations, which makes knowing when not to trust a model's output as important as the output itself. This has driven growing interest in uncertainty quantification (UQ) for LLMs: methods that measure a model's confidence in its generations. Several families of methods have been proposed, including token probabilities-based, output consistency-based, hybrid (combining token probabilities and semantic consistency) and verbalized approaches (Vashurin et al., 2025a; Xia et al., 2025). Despite their diversity, all share a fundamental limitation: reducing uncertainty to a single scalar confidence score.

While such scores are useful as a proxy for the quality of the generation, they provide limited insight into the underlying source of uncertainty. However, in natural language generation uncertainty can arise from fundamentally different sources – limited model knowledge, legitimate output variability, and ambiguous inputs (Baan et al., 2023; Prabhudesai et al., 2024). This raises an important question: *how do existing UQ methods actually behave under different uncertainty sources?*

Meaningful comparison of UQ methods across uncertainty sources requires a dataset carefully designed to introduce each uncertainty source in a controlled way, ideally through variations of the same question so that differences in UQ behavior can be attributed to the uncertainty source rather than to confounding factors such as answer length, format, or task domain. However, existing uncertainty-related datasets fall into two categories: they either introduce ambiguity without separating its sources – as in AmbigQA (Min et al., 2020), or target a single highly specific uncertainty source, such as EverGreenQA (Pletenev et al., 2025), which focuses exclusively on temporal uncertainty.

In this work, we address this gap by constructing a dataset with three types of questions, each designed to target a distinct source of uncertainty, and conducting the first systematic evaluation of UQ methods under controlled uncertainty conditions. Our contributions can thus be summarized as follows:

- A new human-validated dataset consisting of questions organized into triplets, designed to introduce model knowledge, output variability, and input ambiguity uncertainty in a controlled manner.
- We evaluate 16 uncertainty methods across 5 models under controlled uncertainty conditions, finding that performance degrades consistently once uncertainty sources beyond model knowledge limitations are introduced.
- An analysis revealing that output variability and input ambiguity induce opposite failure modes across all method families in ways that are predictable from each family's core assumptions.

## 2 Related Work

**Sources of Uncertainty in NLG.** In machine learning, uncertainty is commonly decomposed into epistemic and aleatoric uncertainty (Hüllermeier and Waegeman, 2021). Epistemic uncertainty stems from gaps in model knowledge and can, in principle, be reduced. Aleatoric uncertainty is considered inherent to the task and is irreducible. However, in natural language generation, uncertainty can arise from a broader variety of sources (Prabhudesai et al., 2024; Baan et al., 2023; Hu et al., 2023). Existing taxonomies largely agree on three high-level origins of uncertainty: input, model, and output. These do not always map cleanly onto the epistemic/aleatoric dichotomy: output variability reflects genuine semantic diversity rather than irreducible noise, and input ambiguity is reducible through clarification (Kirchhof et al., 2025; Baan et al., 2023). In our work, we treat these three sources as operationally distinct, as each corresponds to different model behavior and calls for a different system response.

**Uncertainty Quantification for LLMs.** Recent work on Uncertainty Quantification for LLMs has proposed a diverse set of methods that attempt to estimate the confidence of the model. These methods can be grouped into five main categories: token-probabilities based, consistency-based, hybrid, verbalized, and internal state-based (Vashurin et al., 2025a; Xia et al., 2025). Token-probability-based methods derive uncertainty from the model's predictive distribution; consistency-based methods measure agreement across multiple sampled responses; hybrid methods combine both signals; verbalized methods prompt the model to explicitly state its confidence; and internal-state-based methods use hidden representations to estimate uncertainty. In this work, we focus on the first four families as they are the most widely adopted in practice and require no access to model internals beyond token probabilities and sampled outputs, making them applicable across a broad range of deployments. While reflecting different perspectives on how uncertainty in LLM outputs can be measured, these methods are designed to produce a single scalar uncertainty score. Consequently, existing UQ approaches do not explicitly characterize which underlying sources of uncertainty contribute to a given score, limiting their ability to distinguish between qualitatively different forms of uncertainty. Recent work has begun to expose this limitation: Tomov et al. (2026) show that UQ methods can degrade to near-random performance under aleatoric uncertainty and provide a theoretical account of this failure. We complement and expand upon this by adopting a three-source view of uncertainty, evaluating methods under model knowledge limitations, output variability, and input ambiguity in a controlled, comparable setting.

**Uncertainty-Focused Datasets.** Relatively few datasets explicitly focus on uncertainty in natural language generation. Fewer still deliberately construct questions to isolate different sources of uncertainty. One of the earliest examples, AmbiNQ (Min et al., 2020), introduces a collection of questions that can be interpreted in several ways due to underspecification, missing references, etc. More recent work, MAQA (Yang et al., 2025), deals with questions that have a list of multiple possible answers. Other work targeted more specialized sources of uncertainty. For example, EverGreenQA focuses on uncertainty introduced by temporal dependencies: questions whose correct answer is time-sensitive (Pletenev et al., 2025). Rajpurkar et al. (2018) propose an unanswerable SQuADRUn dataset, which contains 50,000 questions the model should abstain from answering. However, these datasets, remain limited in scope: they focus on a single uncertainty source or do not separate sources, and few – to our knowledge – provide a systematic framework that differentiates between multiple uncertainty types in a controlled, comparable way. We bridge this gap.

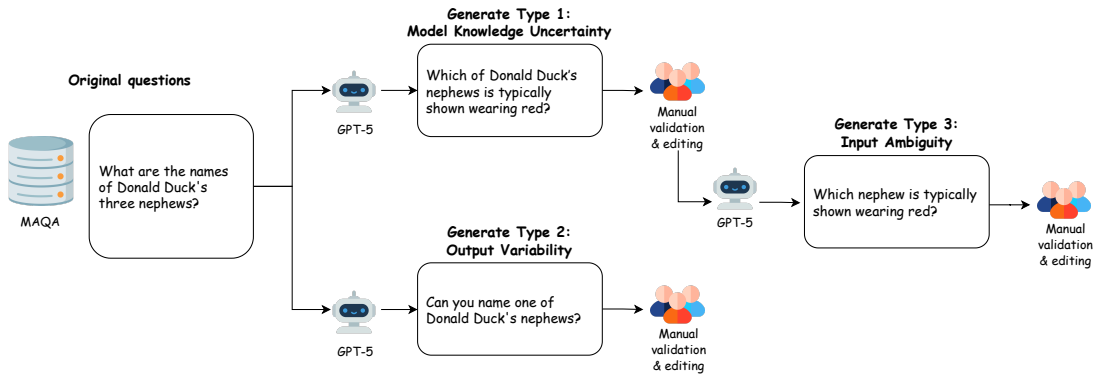



Figure 1: Question generation pipeline used to construct question triplets.

## 3 Dataset Creation

In this section, we describe the proposed dataset design, data sources, and construction procedure.

### 3.1 Choice of Uncertainty Sources

Prior work has proposed a wide range of uncertainty sources in natural language generation, including more specific subcategories like linguistic variability, and uncertainty induced by stochastic decoding. However, many of these proposed uncertainty sources are either difficult to isolate empirically, model-dependent, or tightly correlated. Thus, we intentionally focus on a minimal but operational subset of uncertainty sources that (i) frequently arise in real-world question answering, (ii) can be cleanly separated through controlled dataset construction, and most importantly (iii) correspond to distinct downstream actions a system might take. This motivates our focus on the following three sources of uncertainty: model knowledge uncertainty, output variability, and input ambiguity.

**Model Knowledge Uncertainty.** When the question is well-specified and presupposes a single correct response, we can conclude that uncertainty stems from the limitations of model knowledge. A certain degree of response variability due to inherent language properties (like semantically equivalent expressions, which differ only in particular wording) can still confound the true model uncertainty. However, many modern UQ methods (Kuhn et al., 2023; Duan et al., 2024; Fadeeva et al., 2024) are well-equipped to handle this interference by means of semantic clustering of the model outputs. Operationally, prevalence of this type of uncertainty should result in either abstention from answering or using external knowledge sources to augment the parametric knowledge of the model.

**Output Variability.** A well-specified question can have several semantically distinct plausible answers. For instance, the question *"Give me an example of an animal that lives in the desert."* can be correctly answered by naming any of the desert-dwelling animals. In general setting, it's not obvious why this particular case should result in high uncertainty – any response is as good as the other and can be given with confidence. Within probabilistic model of modern LLMs, however, this will likely result in the spread of the probability mass among candidate responses. For example, Yang et al. (2025) observe empirically that peak answer probability decreases with the number of valid answers while their aggregate remains stable, indicating a spread of probability mass across valid candidates. Drawing outputs from this distribution will result in semantically inconsistent responses. Some of the most prominent families of the UQ methods will treat this signal as high uncertainty.

**Input Ambiguity.** An ambiguous input naturally induces uncertainty in the output. Such ambiguity can be caused by a multitude of reasons. Unresolved references to entities or actors, temporal ambiguities, missing qualifiers, coreference and many other qualities of the question, when present, can make it underspecified and preclude answering it with confidence. Moreover, underspecification complicates further analysis of uncertainty. Indeed, it's much harder to assess if the model knows the answer to a question if it's unclear what the question is actually about. From a decision-making perspective, uncertainty stemming from input ambiguity is best resolved by asking the user to clarify the question.

### 3.2 Dataset Construction

To construct the dataset, we start from a subset of the MAQA dataset (Yang et al., 2025) derived from Natural Questions, containing questions with a list of possible answers. We start from multi-answer questions because it is generally easier to convert a multi-answer question into a single-answer than the other way around. A full dataset construction pipiline can be seen on Figure 1.

**Type 1: Model Knowledge Uncertainty.** To construct questions for this type, we begin with the multi-answer question from MAQA dataset. For each original question, we randomly select one of its valid answers as the target answer. We then prompt GPT-5 to rewrite the question into a single-answer version such that only the selected target answer remains valid . This transformation ensures that the resulting question is well-posed and unambiguous, with a unique correct response corresponding to the chosen target. Next, we conduct human validation of the generated question–answer pairs. Annotators are instructed to verify (1) the correctness and non-ambiguity of each question and (2) the factual accuracy of the corresponding answer. To confirm correctness, annotators perform an independent Google search and cross-check the retrieved evidence. If an answer is found to be incorrect or inconsistent with the question, the annotators manually rewrite the question to align it with the provided target answer. This step ensures both linguistic precision and factual reliability of the single-answer data. Questions that are difficult to convert into a single-answer format are rejected.

**Type 2: Output Variability.** We then construct questions of Type 2. The questions in the MAQA dataset are associated with a list of correct answers and are typically phrased to imply that all valid answers should be provided. However, our investigation requires that a question be answered correctly by naming one valid answer rather than enumerating all possible answers. Thus, we reformulate each question to require only a single correct response, where any one of the valid answers is accepted as correct. For example, original question "What were the names of the two main factions in the English Civil War?" was reformulated into "Name one of the two main factions in the English Civil War." This reformulation is performed using GPT-5 model, followed by human validation. Note that model knowledge uncertainty remains present in this setting: a model may fail to produce any of the valid answers. We can cleanly separate the two sources of uncertainty post-hoc – a generation matching any valid answer reflects output variability as the operative source, while an incorrect generation indicates model knowledge limitations.

**Type 3: Input Ambiguity.** We derive Type 3 questions from the validated Type 1 questions rather than Type 2, to maintain a clear separation between ambiguity in the input and variability in the output. Starting from Type 1 ensures each base question has a unique answer, so any uncertainty in Type 3 can be attributed solely to underspecification. We prompt GPT-5 to remove key contextual information – typically replacing specific referents with vague expressions – introducing referential ambiguity while avoiding overlap with other uncertainty sources. Human annotators then verify that sufficient context has been removed such that the question cannot be answered without clarification, and manually edit cases that remain weakly ambiguous. We reject questions that collapse into generic forms (e.g., "Who said this?") as they lose domain specificity. By design, Type 3 questions are unanswerable without clarification.

Finally, we compile question triplets, discarding any triplet in which a question was rejected at any stage. This yields 539 questions per uncertainty type (1,617 total), with uncertainty sources introduced in a controlled way and consistent structure and domain across all three types. The balanced design ensures fair comparison across uncertainty sources. Prompts and annotator guidelines are provided in Appendix A.

## 4 Experiments

### 4.1 Experimental Setup

We run all experiments through the `lm-polygraph` library (Fadeeva et al., 2023; Vashurin et al., 2025a), which offers standardized implementations of UQ methods, ensuring consistent evaluation across methods.

**Models.** We evaluate 4 open-weight models – Llama-3.1-8b-Instruct, Gemma-3-12b-Instruct, Qwen-2.5-14b-Instruct, and Llama-3.3-70b-Instruct and GPT-4.1 as example of a proprietary model. Model endpoints are detailed in Appendix C.

| Setting | Question Types | Desciption |
|---|---|---|
| **MK** | Type 1 | Well-formed questions with a unique correct answer. Model knowledge is the sole source of uncertainty. |
| **MK + OV** | Type 2 | Questions admitting multiple valid answers. Output variability is introduced alongside model knowledge uncertainty. |
| **MK + IA** | Type 1 + Type 3 | Mix of well-formed and underspecified questions. A well-calibrated method must remain confident on correctly answered Type 1 and assign high uncertainty to Type 3. |
| **MK + OV + IA** | Type 1 + Type 2 + Type 3 | All question types combined, reflecting realistic deployment conditions where all uncertainty sources co-occur. |

Table 1: Evaluation settings. Model knowledge (MK) uncertainty is present in all settings as an inherent property of any deployed system; remaining settings layer output variability (OV) and input ambiguity (IA) on top.

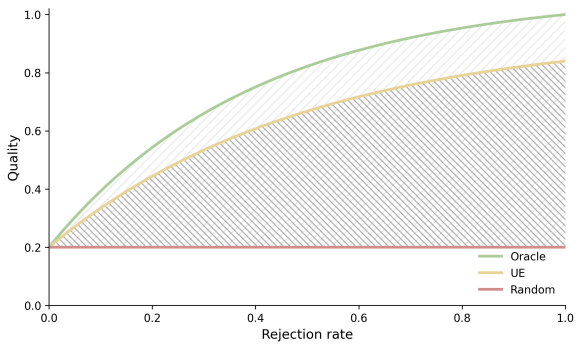



Figure 2: Prediction–Rejection Ratio (PRR) curve of non-rejected prediction quality vs. rejection rate. Oracle denotes the optimal strategy; Random and UQ correspond to random and uncertainty-based rejection.

**Methods.** We evaluate a wide range of methods of different types. First, we include the following token-probabilities-based methods: Sequence Probability (SP), Perplexity (PPL) (Fomicheva et al., 2020), Mean Token Entropy (MTE) (Fomicheva et al., 2020), Monte Carlo (Normalized) Sequence Entropy (MCSE/MCNSE) (Kuhn et al., 2023; Malinin and Gales, 2021). Second, we consider several consistency-based methods: Number of Semantic Sets (NumSemSets), Degree Matrix (DegMat), Sum of Eigenvalues of the Graph Laplacian (EigValLaplacian), and Eccentricity (Lin et al., 2024). We then evaluate state-of-the-art hybrid methods: Semantic Entropy (Kuhn et al., 2023), Semantic Density (Qiu and Miikkulainen, 2024), CoCoA SP, CoCoA PPL, CoCoA MTE (Vashurin et al., 2025b). Lastly, we evaluate verbalized uncertainty: 2S-Verbalized Uncertainty (Tian et al., 2023). Detailed descriptions of the methods used can be found in Appendix D.

**Evaluation Metrics.** We adopt Prediction Rejection Ratio (PRR) as our primary evaluation metric for the task of selective generation, following the previous works (Vashurin et al., 2025a). PRR operates by progressively rejecting predictions based on uncertainty scores and observing how the average quality of the remaining predictions changes (see Figure 2). It is calculated as the ratio of two areas: the area between the Prediction Rejection (PR) curves for the evaluated uncertainty score ($\text{AUC}_{\text{unc}}$) and the area between the oracle ($\text{AUC}_{\text{oracle}}$ ) - the ideal uncertainty score that perfectly ranks instances by quality. These areas are additionally normalized by random rejection baseline ($\text{AUC}_{\text{rnd}}$). Formally, PRR is defined as follows:

$$PRR = \frac{\text{AUC}_{\text{unc}} - \text{AUC}_{\text{rnd}}}{\text{AUC}_{\text{oracle}} - \text{AUC}_{\text{rnd}}}. \quad (1)$$

We limit the rejection to 75%, as rejecting 100% of generation is impractical in real-world applications.

**Quality Metric.** We employ an LLM-as-a-judge (GPT-5) to assess generation quality. For Type 1 and Type 2 questions, the judge labels each response as correct or incorrect with respect to reference answer. For Type 3 questions, it determines whether the model requested clarification (correct) or committed to a specific answer (incorrect). Each generation is assigned a binary score of 1 for correct and 0 for incorrect. Full prompt templates and human validation are in Appendix B.

**Evaluation Settings.** We assess uncertainty quantification methods across 4 settings, summarized in Table 1. The **MK** setting evaluates Type 1 questions in isolation, establishing a baseline where model knowledge uncertainty is the sole active uncertainty source. The **MK + OV** setting

| Metric | MK | MK+OV | MK+IA | MK+OV+IA |
|---|---|---|---|---|
| SP | 0.580 | 0.417 ↓0.163 | 0.438 ↓0.142 | 0.415 ↓0.165 |
| PPL | 0.556 | 0.372 ↓0.184 | 0.415 ↓0.141 | 0.372 ↓0.184 |
| MTE | 0.593 | 0.405 ↓0.188 | 0.445 ↓0.148 | 0.401 ↓0.192 |
| MCSE | 0.587 | 0.425 ↓0.162 | 0.439 ↓0.149 | **0.426 ↓0.161** |
| MCNSE | 0.561 | 0.371 ↓0.190 | 0.400 ↓0.161 | 0.368 ↓0.193 |
| Consistency | 0.511 | 0.386 ↓0.125 | 0.358 ↓0.153 | 0.320 ↓0.191 |
| NumSemSets | 0.448 | 0.302 ↓0.146 | 0.336 ↓0.112 | 0.308 ↓0.140 |
| EigValLaplacian (entail) | 0.580 | 0.420 ↓0.159 | 0.427 ↓0.153 | 0.392 ↓0.187 |
| DegMat (entail) | 0.555 | 0.394 ↓0.160 | 0.372 ↓0.183 | 0.336 ↓0.218 |
| Eccentricity (entail) | 0.553 | 0.415 ↓0.139 | 0.397 ↓0.157 | 0.380 ↓0.173 |
| Semantic Entropy | 0.589 | **0.429 ↓0.161** | 0.444 ↓0.145 | 0.425 ↓0.164 |
| Semantic Density | **0.644** | 0.392 ↓0.252 | 0.427 ↓0.217 | 0.379 ↓0.265 |
| CocoaMSP | 0.596 | 0.422 ↓0.174 | 0.447 ↓0.148 | 0.410 ↓0.186 |
| CocoaPPL | 0.580 | 0.393 ↓0.187 | 0.431 ↓0.149 | 0.383 ↓0.197 |
| CocoaMTE | 0.605 | 0.413 ↓0.192 | **0.453 ↓0.152** | 0.400 ↓0.205 |
| Verbalized2S | 0.611 | 0.262 ↓0.349 | 0.422 ↓0.189 | 0.319 ↓0.292 |

Table 2: PRR scores averaged across all five models. Inline arrows show drop from MK setting. **Bold** and underline denote best and second-best per column.

evaluates Type 2 questions, where output variability is introduced in addition to the model knowledge uncertainty. The **MK + IA** setting jointly evaluates methods on Type 1 and Type 3 questions: since Type 3 questions are unanswerable by construction, they are mixed with Type 1 questions to create a setting where a well-calibrated method must remain confident on correct answers, assign high uncertainty to incorrect ones, and also assign high uncertainty to unanswerable inputs. Finally, the **MK + OV + IA** setting combines all three question types, reflecting the realistic deployment scenario where all sources of uncertainty co-occur and no single source dominates.

### 4.2 Results

**Selective Generation Performance.** Table 2 reports PRR scores averaged across all five models. Detailed experimental results can be found in Appendix F. As can be seen, all methods experience a drop in performance once model knowledge limitation is no longer the only source of uncertainty. In the **MK**-only setting, Semantic Density achieves the highest PRR, followed by Verbalized2S, suggesting that hybrid and verbalized confidence signals are most informative when the input is well-formed and a unique correct answer exists. As shown in Appendix F, Verbalized2S performance in this setting tends to improve with model size, consistent with findings in (Kadavath et al., 2022; Tao et al., 2025).

When output variability is introduced, all methods experience a drop in PRR (MK+OV), indicating that existing approaches struggle to distinguish between a model's knowledge limitations and the multiplicity of valid answers. Hybrid methods remain the best perfoming on average, though the performance gap between them and token-probability- and consitency-based methods narrows compared to the MK-only setting. Verbalized2S suffers the sharpest performance drop, however, the performance is model-dependent: it remains competitive for GPT-4.1, while collapsing for Llama-3.3-70b-Instruct.

Under input ambiguity (MK+IA), all methods again degrade, this time for a different reason: once a model commits to one interpretation of an ambiguous question, it may produce overconfident or internally consistent answers for that interpretation, masking the underlying uncertainty. Hybrid methods again lead, but the gap to pure token-probability methods narrows further. Verbalized confidence remains heavily model-dependent – as detailed in Appendix F, For Llama-70B and Qwen Verbalized confidence is top performering method under this setting, suggesting that some models are better able to recognize and express uncertainty

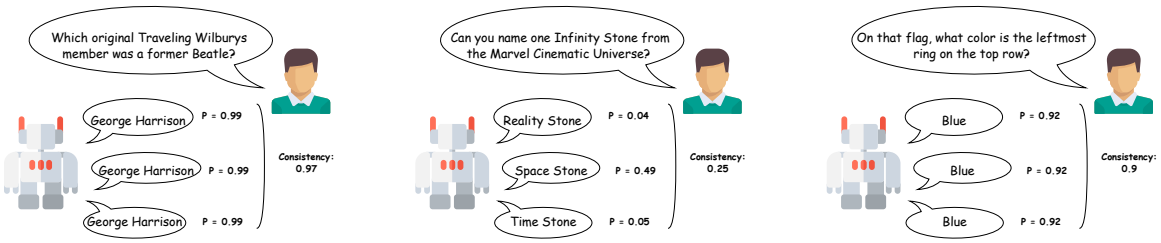



Figure 3: Examples of UQ failure modes across question types. Type 1: consistent, high-probability answers are correct. Type 2: inconsistency reflects valid output diversity, not poor quality. Type 3: the model commits to one interpretation of an ambiguous input, producing falsely confident, consistent answers.

stemming from question underspecification.

In the combined setting (MK+OV+IA), MCSE emerges as the strongest method, with Semantic Entropy close behind. When all sources of uncertainty are present simultaneously, hybrid methods appear to offer diminishing returns for uncertainty quantification performance.

**Model Behavior.** As shown in Table 4, all three models achieve comparable accuracy on single-answer and multi-answer questions, and none reject answering in either setting unprompted. For ambiguous inputs, rather than requesting clarification, models overwhelmingly commit to an arbitrary interpretation and respond with apparent confidence. Generation length distributions are comparable across uncertainty types, confirming that differences in UQ behavior are not artifacts of length variation (Appendix E).

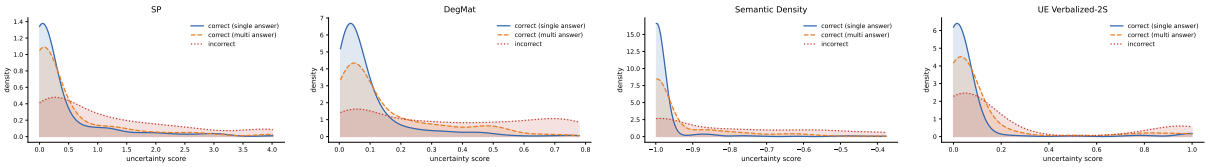
## 5 Discussion

Our results reveal a clear pattern: existing UQ methods underperform when sources of uncertainty apart from model knowledge limitations are introduced. However, output variability and input ambiguity expose qualitatively different failure modes in UQ methods (Figure 3).

**Existing methods are most suited for well-specified, single-answer settings.** When model knowledge is the only source of uncertainty, UQ methods perform at their best. In this setting, both token-probability-based and consistency-based methods align well with generation quality. A model that knows the answer will produce a confident, and self-consistent response, while a model that does not know will assign low probabilities and generate a variety of answers. Because both signals point in the same direction, hybrid methods not only provide the best performance here, but also provide the most gain from uniting signals. Verbalized uncertainty is also competitive here, with performance improving as model size increases - larger models appear better calibrated at introspecting on their own knowledge limitations, with GPT-4.1 achieving a PRR of 0.855 against 0.427 and 0.437 of Llama-3.1-8b-Instruct and Gemma-3-12b-Instruct (see Appendix F).

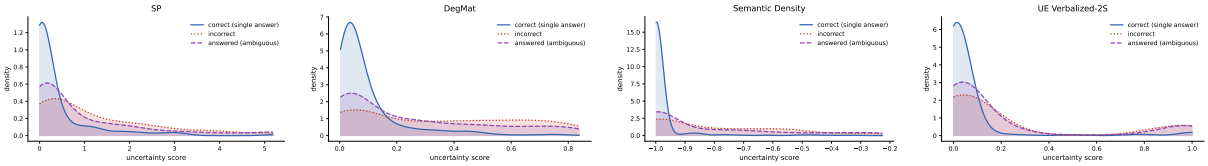
**Output variability inflates uncertainty estimates.** When a question admits multiple valid answers, UE scores increase across all method families (Figure 4, Appendix G). Token-probability–based methods fail here for a principled reason: when an input admits $N$ equally valid completions $\{y_1, \ldots, y_N\}$, probability mass is distributed across all of them, yielding $p(y_i \mid x) = 1/N$ and thus $U_{\text{SP}}(y_i \mid x) = \log N$. Any individual correct completion therefore carries lower probability as the number of valid alternatives grows, and the method misattributes this dilution as a knowledge gap. Consistency-based methods fail for a parallel reason: a well-calibrated model naturally produces diverse valid responses, so low agreement across samples reflects legitimate answer diversity rather than genuine uncertainty (Figure 3). Both families thus penalize the model for doing exactly what the task requires. Hybrid methods, inheriting both signals, compound these effects. Verbalized confidence can also be overstated, though the degree is model-dependent.

**Input ambiguity deflates uncertainty estimates.** When a question is underspecified, the opposite failure occurs: UE scores remain lower than those observed on incorrectly answered questions, masking genuine ambiguity (Figure 5, Appendix G.). Under input ambiguity, models can commit to an arbitrary interpretation and answer




(a) Sequence Probability (b) Degree Matrix (c) Semantic Density (d) Verbalized-2S

Figure 4: Density of UQ scores for correct single- and multi-answer, and incorrect responses. Output variability shifts the correct distribution toward higher uncertainty, increasing overlap with incorrect answers. Model – Gemma-3-12b-Instruct.




(a) Sequence Probability (b) Degree Matrix (c) Semantic Density (d) Verbalized-2S

Figure 5: Density of UQ scores for correct, incorrect answers and answered ambiguous questions. Models commit to an arbitrary interpretation and answer with apparent confidence – uncertainty estimates for ambiguous inputs are lower than for incorrect answers, indicating overconfidence. Model – Gemma-3-12b-Instruct.

with apparent confidence rather than signaling that the question is unclear (Figure 3). This is also supported by the near zero clarification rates (Table 4). Token-probability–based methods consequently assign high likelihood to generic responses compatible with several interpretations simultaneously, returning low uncertainty despite unresolved ambiguity in the input. Consistency-based methods are similarly deceived: a model that collapses onto a single interpretation can produce a highly consistent outputs, so agreement is high and uncertainty appears low. Hybrid methods again inherit failure modes from both. Verbalized methods for some models, can better detect this source of uncertainty suggesting that sufficiently capable models can recognize underspecification even without explicit instruction to do so.

## 6 Conclusion and Future Work

In this paper, we introduced a dataset for controlled evaluation of UQ methods under distinct sources of uncertainty, and conducted a systematic evaluation across 5 models. Our experiments show that existing methods perform well when model knowledge limitation is the sole source of uncertainty, but fail when output variability and input ambiguity are introduced. Crucially, UQ methods fail in different ways for these sources: output variability inflates uncertainty estimates by penalizing legitimate answer diversity, while input ambiguity deflates them as models commit to arbitrary interpretations with apparent confidence.

Our results suggest several directions for improving UQ in LLMs. First, moving beyond scalar confidence scores toward *structured uncertainty representations* that disentangle knowledge uncertainty, output variability, and input ambiguity is a key challenge. Second, future work should focus on *uncertainty-aware decision policies* that map these sources to appropriate actions (e.g., abstention, clarification, or diversification), rather than evaluating UQ in isolation. Third, extending our benchmark to more realistic settings-such as multi-hop reasoning, long-form generation, and multilingual data-would test the robustness of current methods.

## Limitations

Although our work offers insight into the nature of uncertainty sources, it has several limitations.

First, our analysis is restricted to question answering tasks. Although QA provides a controlled setting for isolating and studying different uncertainty sources, uncertainty may manifest differently in other generation tasks such as summarization, translation, or creative writing.

Second, we focus on standard instruction-tuned models in our evaluation and do not include reasoning models. Reasoning models present a fundamentally different setting for UQ, as their extended chain-of-thought generation introduces distinct challenges – such as reasoning path divergence and multi-step error propagation – that are not present in instruction-tuned models (Liu et al., 2025). The UQ methods we evaluate were specifically designed for standard autoregressive generation, and instruction-tuned models remain widely deployed in production environments due to their lower inference cost and latency. Evaluating UQ under reasoning model settings is an important direction we leave for future work.

Third, our dataset prioritizes controlled, human-validated examples and is therefore limited in scale. However, with 539 questions per uncertainty type, the dataset is sufficiently large to detect moderate effect sizes, and the observed differences in UQ method performance are consistent across all three models, suggesting our findings reflect genuine behavioral patterns.

## Ethics Statement

**Intended Use and Scope.** This work is a measurement and evaluation study of uncertainty quantification (UQ) methods for large language models. It does not introduce new model architectures, training procedures, or deployed systems. The goal is to better understand limitations of existing UQ approaches and support safer, more transparent use of LLMs.

**Data Collection and Usage.** Our dataset is constructed from publicly available question–answer sources and synthetic transformations. No personal data, sensitive attributes, or user-specific information are included. The dataset is designed for research purposes only and does not target any specific demographic group. While we apply human validation, we acknowledge that source data and model-generated content may reflect existing societal biases.

**Human Annotation.** Human annotators (graduate students with NLP/ML background) were involved in validating and editing question–answer pairs. Annotation was limited to factual verification and linguistic clarity. Annotators were not exposed to harmful or sensitive content beyond standard factual QA material, and no personal data was handled during the process.

**Risks and Potential Misuse.** A potential risk is that improved or misinterpreted uncertainty estimates could increase user trust in incorrect model outputs. Our findings explicitly show that existing UQ methods can be misleading under output variability and input ambiguity. We therefore caution against using UQ scores as a sole indicator of correctness, especially in high-stakes applications. Uncertainty signals should be combined with additional safeguards such as human oversight, retrieval, or verification systems.

**Limitations and Generalization.** Our evaluation focuses on controlled question answering scenarios with three isolated sources of uncertainty. While this enables systematic analysis, it does not capture the full complexity of real-world interactions. Results may not directly transfer to other tasks (e.g., dialogue, long-form generation) or to proprietary, black-box, or tool-augmented systems.

**Broader Impact.** By identifying failure modes of current UQ methods, this work aims to reduce overconfidence and improve reliability in LLM-based systems. We emphasize that uncertainty should be treated as a structured phenomenon and used to inform cautious and context-aware system behavior, rather than as a standalone signal for automated decision-making.

## References

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, R. Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty in natural language generation: From theory to applications. *ArXiv*, abs/2307.15703.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings*

*of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063. Association for Computational Linguistics.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications. *arXiv preprint arXiv:2306.04459*.

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Michael Kirchhof, Gjergji Kasneci, and Enkelejda Kasneci. 2025. Position: Uncertainty quantification needs reassessment for large language model agents. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6107–6117, New York, NY, USA. Association for Computing Machinery.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Sergey Pletenev, Maria Marina, Nikolay Ivanov, Daria Galimzianova, Nikita Krayko, Mikhail Salnikov, Vasily Konovalov, Alexander Panchenko, and Viktor Moskvoretskii. 2025. Will it still be true tomorrow? multilingual evergreen question classification to improve trustworthy QA. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8614–8631, Suzhou, China. Association for Computational Linguistics.

Snehal Prabhudesai, Daniel G Goldstein, Jake Hofman, and David M Rothschild. 2024. A taxonomy for understanding and identifying uncertainty in AI-generated responses. *SSRN Electron. J.*

Xin Qiu and Risto Miikkulainen. 2024. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Linwei Tao, Yi-Fan Yeh, Minjing Dong, Tao Huang, Jialin Yu, Philip Torr, and Chang Xu. 2025. Revisiting uncertainty estimation and calibration of large language models. In *Workshop on Scaling Environments for Agents*.

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

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

Tim Tomov, Dominik Fuchsgruber, Tom Wollschläger, and Stephan Günnemann. 2026. The illusion of certainty: Uncertainty quantification for llms fails under ambiguity. *Preprint*, arXiv:2511.04418.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. 2025a. Benchmarking uncertainty quantification methods for large language models with LM-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

Roman Vashurin, Maiya Goloburda, Albina Ilina, Aleksandr Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. 2025b. Cocoa: A minimum bayes risk framework bridging confidence and consistency for uncertainty quantification in LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. 2025. A survey of uncertainty estimation methods on large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21381–21396, Vienna, Austria. Association for Computational Linguistics.

Yongjin Yang, Haneul Yoo, and Hwaran Lee. 2025. MAQA: Evaluating uncertainty quantification in LLMs regarding data uncertainty. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5846–5863, Albuquerque, New Mexico. Association for Computational Linguistics.

# A Dataset Construction

## A.1 Prompts

In this section, we provide system prompts and model prompts that are used to rewrite the questions for each scenario.

```
You are a rewriting assistant that refines multi-
answer questions into clear, natural single-answer
questions.
You will be given:
1. An original question that may have multiple
correct answers.
2. A list of possible answers.
3. A specific target answer from that list.
Your goal:
- Rewrite the question so it can only be answered
correctly with the target answer.
- Keep the question fluent, natural, and concise (
preferably under 25 words).
- The result should sound like a well-phrased quiz
or interview question - not an encyclopedia entry.
Guidelines:
- Use discriminative details that uniquely identify
the target (e.g., creator, origin, main feature,
time period, famous example, key achievement).
- Add just enough context to exclude other listed
answers, but avoid long or overly technical
descriptions.
- Do NOT include the target answer itself in the
question.
- Do NOT fabricate or distort facts.
- Maintain the same knowledge domain as the original
 question.
Tone and style:
- Favor natural English phrasing over formal or
taxonomic style.
- Avoid stacked adjectives, nested clauses, or
parentheses.
- Write as if asking a person in conversation: short
, clear, and specific.
Output strictly in JSON:
{
  "target_answer": "<target answer>",
  "rewritten_question": "<natural single-answer
question>"
}
```

Listing 1: System prompt for S1.

```
You are given the following data:
Original question: {question}
List of possible answers:
{answers}
Target answer: {target_answer}
Rewrite the question so that it can only be answered
 correctly with the target answer.
Keep the rewritten question concise, factual, and
natural-sounding - as if it appeared in a quiz or
exam, not a textbook.
Use a distinctive clue (creator, time, feature,
purpose, achievement, or location) that clearly
singles out the target answer from the others.
Avoid unnecessary technical jargon or excessive
qualifiers.
Return only this JSON:
{
  "target_answer": "<target answer>",
  "rewritten_question": "<natural single-answer
question>"
}
```

Listing 2: Model prompt for S1.

```
You are a rewriting assistant that reformulates
broad, multi-answer questions into natural, single-
response questions.
Your task:
Rephrase a question that could have many valid
answers so it instead expects one reasonable example
 - while keeping the same meaning and topic.
Examples:
- "Name countries in Europe" -> "What is a country
in Europe?"
- "List programming languages" -> "What is an
example of a programming language?"
- "Which animals live in the desert?" -> "Can you
name an animal that lives in the desert?"
- "Give examples of renewable energy sources" -> "
What is an example of a renewable energy source?"
Guidelines:
- The rewritten question must remain correct if
answered by ANY of the provided valid answers.
- Make it sound natural and varied - not every
question should start with "Name one...".
- Use phrasing such as "What is...", "Can you give
...", "Which is...", "What's an example of...", etc.
- Keep it concise (under 25 words).
- Do NOT add new information or change the scope.
- Avoid phrases like "exactly one" or "only one".
Output strictly in JSON:
{
  "rewritten_question": "<single-response question>"
}
```

Listing 3: System prompt for S2.

```
You are given the following data:
Original question: {{question}}
List of valid answers: {{answers}}
Rewrite the question so that:\n
- It can be answered with ANY of the listed answers,
- It requests only one response (e.g., "Name one
..."),
- It keeps the same meaning and domain,
- It sounds fluent and natural.
Return only this JSON:\n
{
  "rewritten_question": "<single-response question>"
}
```

Listing 4: Model prompt for S2.

```
You generate realistic pairs of questions that
illustrate genuine ambiguity and its resolution.
Each output pair must include:
1) An ambiguous, natural-sounding WH-question (Who/
What/When/Where/Which/How) that cannot be answered
correctly without extra context.
2) A disambiguation question - the minimal, specific
 WH-question that restores the missing info.
Quality constraints:
- Keep both questions <=25 words, fluent, and
conversational.
- Preserve the same domain/topic and answer type.
- Create ambiguity by removing exactly critical
piece of info (referent, time, place, number,
condition, or cause).
- Anchor the question with at least one concrete
noun phrase from the input (e.g., competition name,
country, team, era, instrument, field).
- Do not produce low-content templates (e.g., "Who
said that in that year?", "Which team did that then
?").
- Avoid stacked deictics ("that... that...") and
vague placeholders ("that thing," "those people")
unless there is a clear, single missing slot.
- Do not introduce new facts or entities.
- The disambiguation question must directly request
the missing slot (e.g., "Which year do you mean?", "
Which club?"), not meta-questions like "What do you
mean?".
Output strictly as JSON:
{
  "rewritten_question": "<ambiguous question>",
  "target_answer": "<clarification question>"
}
Examples:
Input:
Question: Which club became the first to win the
European Cup, Cup Winners' Cup, and UEFA Cup?
Answer: Juventus
Output:
{
  "rewritten_question": "Which club was the first to
 win all three of those tournaments in UEFA
competition?",
  "target_answer": "Which three UEFA tournaments are
 you referring to?"
}
Input:
Question: What year was that building built?
Answer: 1979
Output:
{
  "rewritten_question": "When was that building
constructed?",
  "target_answer": "Which building are you referring
 to?"
}
Input:
Question: Which Indo-Aryan language, primarily
spoken in the Jammu region and added to the Eighth
Schedule in 2003, is one of India's officially
recognized languages?
Answer: Dogri
Output:
{
  "rewritten_question": "Which Indo-Aryan language,
mainly spoken in that region, is officially
recognized in India?",
  "target_answer": "Which region do you mean?"
}
```

Listing 5: System prompt for S3.

```
You are given the following data:
Question: {{question}}
Answer: {{target_answer}}
Create one natural-sounding ambiguous WH-question
and its corresponding disambiguation question.
The ambiguous question should stay within a single
domain (e.g., history, sports, geography, science)
and use one or more ambiguity types: referential,
temporal, locational, quantitative, causal, or
contextual.
Make the phrasing fluent and human - avoid
mechanical patterns like 'based in that city'.
Keep both questions concise (<=25 words) and clearly
 related.
Return ONLY this JSON:
{
"rewritten_question": "<ambiguous question>",
"target_answer": "<clarification question>"
}
```

Listing 6: Model prompt for S3.

### A.2 Annotator Guidelines

In this section, we provide the annotator guidelines used for each question type. All annotations were conducted in-house by graduate students specializing in Machine Learning and Natural Language Processing, each proficient in English. Annotation was performed using Google Sheets.

**Type 1: Model Knowledge Limitations.** For this scenario, we construct unambiguous questions with single possible answer. The data is generated by GPT-5 model and verified by human annotators. Out of 592 questions, 497 were accepted, 67 were accepted with edits, and 28 questions were rejected.

**Annotation Guidelines (Type 1: Model Knowledge Limitations.)** Annotators were instructed to ensure that each question–answer pair is clear, unambiguous, and factually correct. For each pair, annotators first verified that the question is grammatical, understandable, and admits only one valid interpretation and answer. Answer correctness was verified via Google Search. If a question was found to be incorrect or ambiguous, annotators were asked to edit the question (not the answer) minimally so that the given answer becomes correct and the question is well-posed. Each question was assigned one of three statuses: *accepted* (correct as-is), *accepted with edits* (correct after minimal edits), or *rejected* (impossible to make single-answer or too unclear to fix).

**Type 2: Output Variability.** We construct unambiguous questions with multiple possible answer. The data is generated by GPT-5 model and verified by human annotators. Out of 592 questions, 543 were accepted, 48 were accepted with edits, and 1 question was rejected.

**Annotation Guidelines (Type 2: Output Variability.)** Annotators were instructed to ensure that each rewritten question is grammatically correct, fluent, and consistent with the original list of valid answers. Annotators verified that the question is natural-sounding and maintains a conversational tone (e.g., "Name one...", "Give an example of..."), preserves the meaning of the original multi-answer question, and remains valid when answered by *any* of the listed possible answers. Each question was assigned one of two statuses: *accepted* (correct as-is) or *accepted with edits* (correct after minimal edits).

**Type 3: Input Ambiguity.** We construct a set of ambiguous questions for scenario 3 that require the model to ask a clarification question. The initial data was generated using GPT-5 and subsequently verified by human annotators. Out of 592 questions, 348 were accepted as-is, 218 required edits, and 26 were rejected entirely. Overall, we found that the model performs poorly on this task. There is a clear trade-off when introducing instructions: providing too much freedom leads the model to remove excessive contextual information, often collapsing many questions into generic forms such as "Who said that?" or "Who did that?". Conversely, restricting the degree of information removal results in questions that retain too much context or in clarification questions that are confusing or refer to entities that no longer appear in the ambiguous version. As a result, many questions had to be substantially rewritten by human annotators to preserve as much of the question triplets as possible.

**Type 3: Input Ambiguity** Annotators were instructed to ensure that each rewritten question pair correctly introduces ambiguity by removing necessary context, and that the provided clarification question is natural and sufficient to resolve it. For the ambiguous question, annotators verified that a key piece of information is omitted such that the question cannot be answered without clarification, while remaining grammatical, fluent, and within the same domain as the original. Trivial ambiguity (e.g., "Who said that?") was not accepted. For

the clarification question, annotators verified that it is a specific WH-question that directly resolves the ambiguity (e.g., "Which year do you mean?") without introducing new information, and that the original question becomes fully answerable once the clarification is provided. Both questions were required to be natural-sounding and concise ($\leq$25 words), avoiding mechanical phrasing such as "based in that city". Each pair was assigned one of three statuses: *accepted* (meets all criteria as-is), *accepted with edits* (acceptable after minimal edits), or *rejected* (original question too short or generic to introduce meaningful ambiguity without degenerating into empty templates).

## B LLM-as-a-judge Validation

In this section, we provide details on LLM-as-Judge implementation and validation.

```
You are grading the model answer to a SINGLE-ANSWER
factual question.

Choose exactly one label for model answer:

- refused
  The model explicitly refused, declined, or said it
 cannot answer.

- correct
  The model clearly gives the correct answer (or an
unambiguous synonym/alias of it),
  and does NOT also present a different conflicting
answer.

- incorrect
  The model attempts an answer, but:
  - the correct answer (or any reasonable synonym)
does NOT appear, OR
  - it presents multiple different candidates (e.g.
"X or Y", "maybe X or Y")
    where not all candidates are the correct one.

Grading rules:
- Focus on the final answer content, ignore
explanation and reasoning.
- Ignore differences in capitalization, punctuation,
 and minor wording changes.
- Extra descriptive context is fine as long as the
main answer is correct and not contradicted.

Return ONLY the label.

Question:
{original_question}

Model answer:
{model_answer}

Correct answer:
{target_answer}
```

Listing 7: LLM-as-a-Judge prompt for S1

```
You are grading the model answer to a MULTI-ANSWER
factual question.

Select one of the following labels for model answer:
- refused: the model clearly refused or declined to
answer.
- correct: every distinct answer item the model
gives is part of the correct answer set
           (a subset is allowed; the model may give
only  one of the correct items).
- incorrect: at least one distinct answer item is
NOT in the correct answer set.

Grading rules:
- Treat the model's answer as a SET of distinct
items it is clearly presenting as answers.
- The model may give only one item or multiple items
.
- Missing items from the correct set are allowed and
 still count as "correct"
  as long as every given item is in the correct set.
- Ignore order and duplicates.
- Minor wording differences, capitalization, and
simple aliases/synonyms are allowed
  (e.g. "Bob Dylan" vs "Robert Zimmerman").
- If the model just says it doesn't know or refuses,
 use "refused".

Return ONLY the label.

Question:
{original_question}

Model answer:
{model_answer}

Correct answer set:
{target_answer}
```

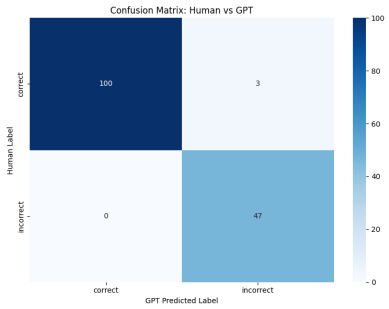
Listing 8: LLM-as-a-Judge prompt for S2

```
You are grading the model's response to an AMBIGUOUS
 user input.

Choose exactly one label for the model response:

- clarification
  The model asks for more information, points out
the ambiguity, or refuses
  to answer. It does NOT commit to a specific
factual answer.

- answered
  The model chooses a specific interpretation of the
 input and attempts to answer it, commiting to a
specific factual answer.

Grading rule:
- If the model both asks for clarification AND
clearly gives a concrete answer,
  label it as "answered".

Return ONLY the label.

User input:
{original_question}

Model response:
{model_answer}
```

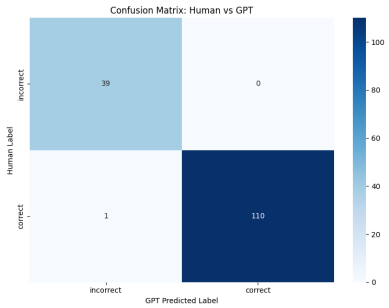
Listing 9: LLM-as-a-Judge prompt for S3

To validate the use of an LLM-as-Judge, we sampled 50 generations from each model for each question type and asked a human annotator to label them following the same evaluation guidelines. As shown in Figure 6, the LLM-as-Judge aligns closely with human judgments, achieving 98.0%, 99.3%, and 100% agreement for single-answer,

multi-answer, and ambiguous questions, respectively. Cohen's k further confirms close agreement between the human annotator and the LLM-as-Judge.




(a) Single-answer




(b) Multi-answer

(c) Ambiguous questions

Figure 6: Agreement between human annotations and LLM-as-Judge across question types.

# C Model Details

```
Answer the following question in ONE sentence.
Answer as plain text on a single line. No line
breaks or markdown. Provide only the answer - no
introductions, explanations, or extra text. No words
 like "here is", "breakdown", "let's", "fascinating
". Keep the answer short. Question: {text}
```

Listing 10: Prompt used to generate answers to questions.

# D Detailed Description of UQ Methods

In this section, we provide detailed description of Uncertainty Quantification methods evaluated in the study.

## D.1 Token-Probabilities Based Methods

**Sequence Probability** is one of the simplest uncertainty measures, defining uncertainty as the negative log-likelihood of the output sequence conditioned on the input.

$$U_{\text{SP}}(\mathbf{y} \mid \mathbf{x}) = -\log p(\mathbf{y} \mid \mathbf{x}). \tag{2}$$

**Perplexity** normalizes the negative log-likelihood by the sequence length, yielding an average per-token uncertainty.

$$U_{\text{PPL}}(\mathbf{y} \mid \mathbf{x}) = -\frac{1}{L} \log p(\mathbf{y} \mid \mathbf{x}). \tag{3}$$

**Mean Token Entropy** computes the average entropy of the model's predictive distribution at each decoding step.

$$U_{\mathcal{H}_T}(\mathbf{y} \mid \mathbf{x}) = \frac{1}{L} \sum_{l=1}^{L} \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}), \tag{4}$$

**Monte Carlo (Normalized) Sequence Entropy** estimates sequence-level uncertainty by averaging the negative log-likelihood over multiple sampled output sequences. This estimate can be further normalized by the length of each sampled sequence.

$$U_{\text{MCSE}}(\mathbf{x}) = -\frac{1}{M} \sum_{i=1}^{M} \log P(\mathbf{y}^{(i)} \mid \mathbf{x}). \tag{5}$$

$$U_{\text{MCNSE}}(\mathbf{x}) = -\frac{1}{M} \sum_{i=1}^{M} \frac{1}{L^{(i)}} \log P(\mathbf{y}^{(i)} \mid \mathbf{x}). \tag{6}$$

## D.2 Consistency-Based Methods

**Number of Semantic Sets (NumSemSets)** quantifies uncertainty as the number of distinct meanings in sampled outputs for a given input, where meanings are clustered using a natural language inference (NLI) model.

**Sum of Eigen Values of Graph Laplacian (EigValLaplacian)** constructs a similarity matrix over sampled outputs and deriving a continuous

| Model | HuggingFace Endpoint |
|---|---|
| Llama-3.1-8b-Instruct | `meta-llama/Llama-3.1-8B-Instruct` |
| Gemma-3-12b-Instruct | `google/gemma-3-12b-it` |
| Qwen-2.5-14b-Instruct | `Qwen/Qwen2.5-14B-Instruct` |
| llamaseventy | `meta-llama/Llama-3.3-70B-Instruct` |

Table 3: HuggingFace Inference Endpoints used in our experiments.

uncertainty score from the eigenvalues of its normalized graph Laplacian.

$$U_{\text{EigV}}(\mathbf{x}) = \sum_{i=1}^{M} \max(0, 1 - \lambda_i(\mathbf{x})). \quad (7)$$

**Degree Matrix** computes uncertainty from the degree matrix $D(\mathbf{x})$ which captures the total similarity of each sampled answer to all others. The diagonal matrix $D$ is defined with entries $D_{ii} = \sum_{j=1}^{M} s_{ij}$. The final score is then defined as:

$$U_{\text{DegMat}}(\mathbf{x}) = 1 - \text{trace}(D(\mathbf{x}))/M^2. \quad (8)$$

**Eccentricity** computes uncertainty by embedding sampled answers into a low-dimensional space using the graph Laplacian, capturing their relational structure based on pairwise similarities. Let $L$ denote the graph Laplacian constructed from the similarity matrix. Let $\mathbf{u}_1, \ldots, \mathbf{u}_K$ be the eigenvectors corresponding to the $K$ smallest eigenvalues of $L$. The embedding of answer $y_j$ is defined as $\mathbf{v}_j = [u_{1,j}, \ldots, u_{K,j}]$. Uncertainty is measured using *eccentricity*, defined as the distance from the center of the embedding space:

$$U_{\text{Ecc}}(\mathbf{x}) = \left\| \mathbf{v}_j - \frac{1}{K} \sum_{\ell=1}^{K} \mathbf{v}_\ell \right\|_2. \quad (9)$$

**Consistency** is a method introduced by  and further utilized by . It is computed using the similarity matrix $S$, where each entry $s_{ij}$ denotes the similarity between sequences $i$ and $j$. The uncertainty score for a sequence $i$ is defined as the average dissimilarity to all other sequences, excluding self-similarity:

$$U_{\text{Cons}}(x_i) = \frac{1}{M-1} \sum_{\substack{j=1 \\ j \neq i}}^{M} (1 - s_{ij}). \quad (10)$$

### D.3 Hybrid Methods

**Semantic Entropy** mitigates the effect of probability discrepancies among semantically equivalent generations. The method groups generated sequences $\mathbf{y}^{(i)}$, $i = 1, \ldots, M$ into semantically coherent clusters $\mathcal{C}_k$, $k = 1, \ldots, K$ using bidirectional entailment. Sequence probabilities are aggregated within each cluster, and uncertainty is computed as:

$$U_{\text{SE}}(\mathbf{x}) = -\sum_{k=1}^{K} \frac{|\mathcal{C}_k|}{M} \log \hat{p}_k(\mathbf{x}), \quad (11)$$

where $\hat{p}_k(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{C}_k} p(\mathbf{y} \mid \mathbf{x})$.

**Semantic Density** measures confidence as the probability mass concentrated near the greedy answer in *semantic space*. Given $M$ sampled generations $\{\mathbf{y}^{(i)}\}_{i=1}^{M}$, we compute a length-normalized probability $\tilde{p}_i = \exp\left(\frac{1}{|\mathbf{y}^{(i)}|} \log p(\mathbf{y}^{(i)} \mid \mathbf{x})\right)$. For each unique sample, a semantic distance to the greedy output is derived from an entailment-based similarity matrix as $d_i = c_i + \frac{1}{2} n_i$ (contradiction $c_i$ and neutral $n_i$). A triangular kernel converts distance to a proximity weight $k_i = \max(0, 1 - d_i)$. The semantic density is then the normalized weighted average

$$\text{SD}(\mathbf{x}) = \frac{\tilde{p}_g + \sum_{i \in \mathcal{U}} \tilde{p}_i \, k_i}{\tilde{p}_g + \sum_{i \in \mathcal{U}} \tilde{p}_i}, \quad (12)$$

$$U_{\text{SemDen}}(\mathbf{x}) = -\,\text{SD}(\mathbf{x}), \quad (13)$$

where $\mathcal{U}$ indexes unique sampled texts and $\tilde{p}_g$ is the length-normalized probability of the greedy generation.

**CoCoA** is a method that combines a token-probability-based uncertainty signal with semantic consistency. Given a target generation $\mathbf{y}_*$ and $M$ sampled generations, the uncertainty estimate is defined as:

| Model | Single-answer | Multi-answer | Clarification rate |
|---|---|---|---|
| Llama-3.1-8b-Instruct | 68.6% (370/539) | 76.3% (411/539) | 3.5% (19/539) |
| Gemma-3-12b-Instruct | 69.0% (372/539) | 76.6% (413/539) | 0.0% (0/539) |
| Qwen-2.5-14b-Instruct | 64.7% (349/539) | 70.5% (380/539) | 0.7% (4/539) |
| Llama-3.3-70b-Instruct | 88.5% (477/539) | 84.6% (456/539) | 0.0% (0/539) |
| GPT-4.1 | 94.8% (511/539) | 91.8% (495/539) | 0.0% (0/539) |

Table 4: Quality of generations across models. For single- and multi-answer questions we report accuracy; for ambiguous inputs we report the clarification rate (fraction of outputs that explicitly ask for clarification).

$$\widehat{U}_{\text{CoCoA}}(\mathbf{y}_* \mid \mathbf{x}) = u(\mathbf{y}_* \mid \mathbf{x}) \cdot \frac{1}{M} \sum_{i=1}^{M} (1 - s_{*i}) = u(\mathbf{y}_* \mid \mathbf{x}) \cdot \widehat{U}_{\text{cons}}(\mathbf{y}_* \mid \mathbf{x}). \quad (14)$$

where $u(\mathbf{y}_* \mid \mathbf{x})$ can be any base uncertainty measure. Following the original formulation, we consider sequence probability, perplexity, and mean token entropy as the underlying uncertainty signals.

### D.4 Verbalized Methods

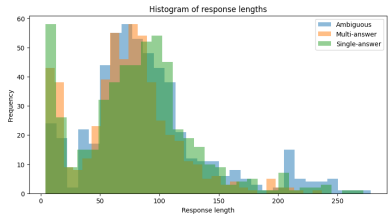
**Verbalized-2S** produces a verbalized confidence score for an already generated response, enabling direct comparison across methods on identical generations. The method re-feeds the original prompt and the generated answer to the model and asks it to explicitly estimate the probability that the answer is correct. In our experiments, we use the following prompt:

> **The question is:**
> {question}
>
> **YOUR GUESS:**
> {candidate_answer}
>
> **Now:**
> Provide the probability that your guess is correct. Give **ONLY** the probability, no other words or explanation.
>
> **For example:**
> Probability: <the probability between 0.0 and 1.0 that your guess is correct >

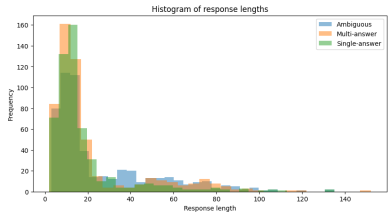
## E Generation Details

In this section, we provide details about the lengths and quality of the generated outputs.

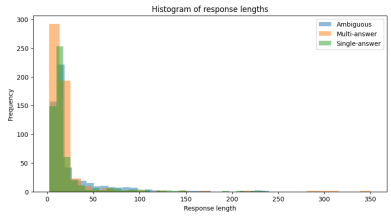
**Lengths.** Since generation length can influence uncertainty quantification, it was important to validate that the produced generations are similarly distributed in terms of length. As shown in Figure 7, all three scenario types exhibit comparable length distributions for each model, meaning that the differences in UE distributions are not caused merely by differences in lengths. However, the distributions differ across models, with Llama-3.1-8b-Instruct producing the longest generations on average and Qwen-2.5-14b-Instruct the shortest.




(a) Llama-3.1-8b-Instruct




(b) Gemma-3-12b-Instruct




(c) Qwen-2.5-14b-Instruct

Figure 7: Distributions of generation lengths across models.

**Quality.** As shown in Table 4, when using LLM-as-Judge as a quality assessor, we observe that none of the models reject answering in the single-

or multi-answer scenarios when the prompt does not explicitly instruct them to do so. For ambiguous inputs, Gemma always provides an answer regardless of the ambiguity, whereas Qwen and LLaMA exhibit only very low clarification rates. Although such behavior can be improved through prompt engineering, our goal in this study is to evaluate uncertainty metrics under different sources of uncertainty; therefore, we intentionally focus on uninstructed generations.

# F Detailed Experimental Results

| Metric | Single | Ambi | Multi | Combined |
|---|---|---|---|---|
| SP | 0.594 | 0.503 ↓0.091 | 0.585 ↓0.009 | 0.501 ↓0.093 |
| PPL | 0.503 | 0.434 ↓0.069 | 0.457 ↓0.046 | 0.413 ↓0.090 |
| MTE | 0.568 | 0.492 ↓0.076 | 0.541 ↓0.027 | 0.485 ↓0.083 |
| MCSE | 0.608 | 0.522 ↓0.086 | **0.643 ↑0.035** | 0.543 ↓0.065 |
| MCNSE | 0.524 | 0.472 ↓0.052 | 0.464 ↓0.060 | 0.443 ↓0.081 |
| Consistency | 0.490 | 0.419 ↓0.072 | 0.479 ↓0.012 | 0.384 ↓0.107 |
| NumSemSets | 0.528 | 0.467 ↓0.061 | 0.584 ↑0.056 | 0.475 ↓0.054 |
| EigValLaplacian | 0.680 | 0.591 ↓0.089 | 0.614 ↓0.066 | **0.546 ↓0.134** |
| DegMat | 0.639 | 0.550 ↓0.088 | 0.595 ↓0.044 | 0.520 ↓0.119 |
| Eccentricity | 0.624 | 0.534 ↓0.090 | 0.610 ↓0.014 | 0.530 ↓0.094 |
| Semantic Entropy | 0.611 | 0.530 ↓0.082 | 0.640 ↑0.029 | 0.544 ↓0.067 |
| Semantic Density | **0.741** | **0.660 ↓0.081** | 0.528 ↓0.213 | 0.544 ↓0.197 |
| CocoaMSP | 0.589 | 0.506 ↓0.083 | 0.587 ↓0.002 | 0.483 ↓0.106 |
| CocoaPPL | 0.519 | 0.452 ↓0.067 | 0.511 ↓0.008 | 0.424 ↓0.095 |
| CocoaMTE | 0.551 | 0.491 ↓0.060 | 0.556 ↑0.005 | 0.465 ↓0.086 |
| Verbalized2S | 0.427 | 0.333 ↓0.094 | 0.201 ↓0.226 | 0.236 ↓0.191 |

Table 5: PRR scores for Llama-8b-it. Inline arrows show drop from Single.

| Metric | Single | Ambi | Multi | Combined |
|---|---|---|---|---|
| SP | 0.564 | 0.394 ↓0.170 | 0.461 ↓0.103 | 0.358 ↓0.206 |
| PPL | 0.544 | 0.357 ↓0.187 | 0.456 ↓0.088 | 0.325 ↓0.219 |
| MTE | 0.587 | 0.378 ↓0.209 | 0.499 ↓0.089 | 0.346 ↓0.242 |
| MCSE | 0.587 | 0.413 ↓0.174 | 0.471 ↓0.117 | 0.374 ↓0.213 |
| MCNSE | 0.551 | 0.355 ↓0.197 | 0.467 ↓0.085 | 0.330 ↓0.221 |
| Consistency | 0.519 | 0.394 ↓0.125 | 0.424 ↓0.095 | 0.328 ↓0.191 |
| NumSemSets | 0.388 | 0.374 ↓0.014 | 0.383 ↓0.005 | 0.324 ↓0.063 |
| EigValLaplacian | 0.531 | 0.402 ↓0.129 | 0.473 ↓0.058 | 0.361 ↓0.170 |
| DegMat | 0.486 | 0.343 ↓0.143 | 0.420 ↓0.066 | 0.289 ↓0.197 |
| Eccentricity (entail) | 0.510 | 0.391 ↓0.119 | 0.453 ↓0.056 | 0.349 ↓0.160 |
| Semantic Entropy | 0.605 | **0.434 ↓0.171** | 0.484 ↓0.120 | **0.386 ↓0.219** |
| Semantic Density | 0.589 | 0.356 ↓0.234 | 0.401 ↓0.188 | 0.320 ↓0.270 |
| CocoaMSP | 0.603 | 0.425 ↓0.178 | 0.483 ↓0.120 | 0.376 ↓0.227 |
| CocoaPPL | 0.591 | 0.404 ↓0.187 | 0.480 ↓0.110 | 0.356 ↓0.235 |
| CocoaMTE | **0.623** | 0.428 ↓0.195 | **0.504 ↓0.119** | 0.373 ↓0.250 |
| Verbalized2S | 0.437 | 0.415 ↓0.022 | 0.338 ↓0.099 | 0.312 ↓0.125 |

Table 6: PRR scores for Gemma-12b-it. Inline arrows show drop from Single.

| Metric | Single | Ambi | Multi | Combined |
|---|---|---|---|---|
| SP | 0.539 | 0.395 ↓0.145 | 0.550 ↑0.010 | 0.394 ↓0.145 |
| PPL | 0.544 | 0.392 ↓0.152 | 0.524 ↓0.020 | 0.371 ↓0.173 |
| MTE | 0.570 | 0.423 ↓0.146 | **0.568 ↓0.001** | 0.408 ↓0.162 |
| MCSE | 0.520 | 0.381 ↓0.140 | 0.531 ↑0.010 | 0.381 ↓0.140 |
| MCNSE | 0.524 | 0.380 ↓0.144 | 0.519 ↓0.005 | 0.358 ↓0.166 |
| Consistency | 0.467 | 0.327 ↓0.140 | 0.498 ↑0.031 | 0.329 ↓0.138 |
| NumSemSets | 0.448 | 0.338 ↓0.111 | 0.354 ↓0.094 | 0.303 ↓0.146 |
| EigValLaplacian | 0.433 | 0.344 ↓0.090 | 0.489 ↑0.056 | 0.342 ↓0.092 |
| DegMat | 0.424 | 0.318 ↓0.106 | 0.485 ↑0.061 | 0.312 ↓0.112 |
| Eccentricity | 0.414 | 0.301 ↓0.113 | 0.478 ↑0.064 | 0.307 ↓0.107 |
| Semantic Entropy | 0.533 | 0.392 ↓0.141 | 0.530 ↓0.003 | 0.386 ↓0.147 |
| Semantic Density | 0.470 | 0.316 ↓0.155 | 0.490 ↑0.020 | 0.337 ↓0.134 |
| CocoaMSP | 0.561 | 0.419 ↓0.143 | 0.548 ↓0.014 | 0.403 ↓0.158 |
| CocoaPPL | 0.560 | 0.417 ↓0.144 | 0.536 ↓0.024 | 0.390 ↓0.171 |
| CocoaMTE | **0.584** | 0.442 ↓0.142 | 0.566 ↓0.018 | **0.416 ↓0.168** |
| Verbalized2S | 0.572 | **0.528 ↓0.044** | 0.400 ↓0.171 | 0.382 ↓0.190 |

Table 7: PRR scores for Qwen-14b-it. Inline arrows show drop from Single.

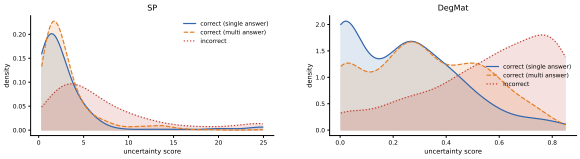
| Metric | Single | Ambi | Multi | Combined |
|---|---|---|---|---|
| SP | 0.707 | 0.476 ↓0.231 | 0.206 ↓0.501 | 0.428 ↓0.279 |
| PPL | 0.680 | 0.483 ↓0.197 | 0.213 ↓0.467 | 0.409 ↓0.271 |
| MTE | 0.702 | 0.514 ↓0.188 | 0.213 ↓0.489 | 0.423 ↓0.279 |
| MCSE | 0.751 | 0.490 ↓0.261 | 0.240 ↓0.511 | **0.447 ↓0.304** |
| MCNSE | 0.704 | 0.446 ↓0.258 | 0.222 ↓0.482 | 0.385 ↓0.318 |
| Consistency | 0.682 | 0.376 ↓0.305 | 0.290 ↓0.392 | 0.311 ↓0.371 |
| NumSemSets | 0.585 | 0.279 ↓0.306 | 0.147 ↓0.437 | 0.237 ↓0.348 |
| EigValLaplacian | 0.786 | 0.443 ↓0.343 | 0.305 ↓0.481 | 0.395 ↓0.391 |
| DegMat | 0.757 | 0.356 ↓0.401 | 0.306 ↓0.451 | 0.314 ↓0.443 |
| Eccentricity | 0.762 | 0.414 ↓0.348 | 0.260 ↓0.502 | 0.371 ↓0.391 |
| Semantic Entropy | 0.743 | 0.478 ↓0.265 | 0.237 ↓0.506 | 0.436 ↓0.308 |
| Semantic Density | **0.814** | 0.442 ↓0.371 | **0.320 ↓0.494** | 0.378 ↓0.436 |
| CocoaMSP | 0.707 | 0.463 ↓0.244 | 0.197 ↓0.511 | 0.406 ↓0.302 |
| CocoaPPL | 0.701 | 0.468 ↓0.232 | 0.208 ↓0.493 | 0.395 ↓0.306 |
| CocoaMTE | 0.718 | 0.484 ↓0.234 | 0.210 ↓0.508 | 0.400 ↓0.317 |
| Verbalized2S | 0.764 | **0.530 ↓0.233** | 0.076 ↓0.688 | 0.347 ↓0.417 |

Table 8: PRR scores for Llama-70b. Inline arrows show drop from Single.

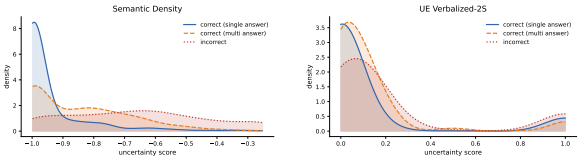
| Metric | Single | Ambi | Multi | Combined |
|---|---|---|---|---|
| SP | 0.495 | 0.424 ↓0.071 | 0.284 ↓0.210 | **0.392 ↓0.102** |
| PPL | 0.510 | 0.409 ↓0.101 | 0.209 ↓0.302 | 0.342 ↓0.168 |
| MTE | 0.537 | 0.419 ↓0.118 | 0.202 ↓0.335 | 0.341 ↓0.196 |
| MCSE | 0.470 | 0.388 ↓0.082 | 0.242 ↓0.228 | 0.386 ↓0.084 |
| MCNSE | 0.503 | 0.347 ↓0.156 | 0.185 ↓0.318 | 0.325 ↓0.178 |
| Consistency | 0.395 | 0.273 ↓0.122 | 0.237 ↓0.157 | 0.246 ↓0.149 |
| NumSemSets | 0.291 | 0.222 ↓0.069 | 0.041 ↓0.250 | 0.203 ↓0.088 |
| EigValLaplacian (entail) | 0.468 | 0.355 ↓0.113 | 0.220 ↓0.248 | 0.317 ↓0.151 |
| DegMat (entail) | 0.468 | 0.291 ↓0.177 | 0.167 ↓0.302 | 0.246 ↓0.222 |
| Eccentricity (entail) | 0.457 | 0.344 ↓0.114 | 0.272 ↓0.186 | 0.345 ↓0.113 |
| Semantic Entropy | 0.454 | 0.389 ↓0.065 | 0.252 ↓0.202 | 0.375 ↓0.079 |
| Semantic Density | 0.605 | 0.362 ↓0.243 | 0.220 ↓0.385 | 0.315 ↓0.290 |
| CocoaMSP | 0.519 | **0.425 ↓0.094** | **0.296 ↓0.223** | 0.384 ↓0.135 |
| CocoaPPL | 0.530 | 0.415 ↓0.115 | 0.230 ↓0.299 | 0.349 ↓0.180 |
| CocoaMTE | 0.551 | 0.421 ↓0.130 | 0.231 ↓0.319 | 0.348 ↓0.202 |
| Verbalized2S | **0.855** | 0.305 ↓0.550 | 0.295 ↓0.560 | 0.320 ↓0.535 |

Table 9: PRR scores for GPT-4.1. Inline arrows show drop from Single.

# G Uncertainty Score Distributions

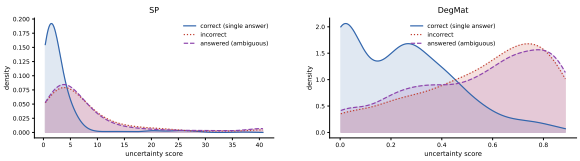



(a) Sequence Probability (b) Degree Matrix

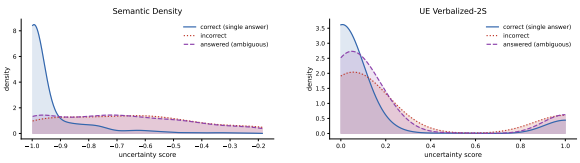



(c) Semantic Density (d) Verbalized-2S

Figure 8: Density of uncertainty scores for correct (single answer), correct (multi answer), and incorrect (pooled) responses. Model - Llama-3.1-8b-Instruct.

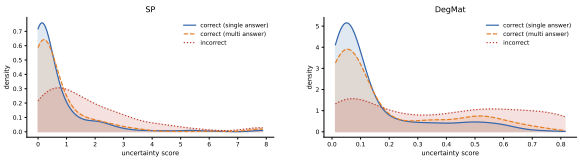



(a) Sequence Probability (b) Degree Matrix

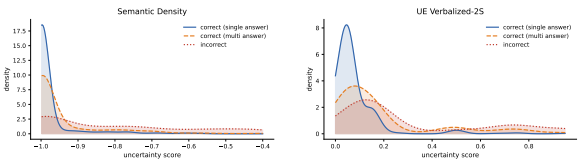



(c) Semantic Density (d) Verbalized-2S

Figure 9: Density of uncertainty scores for correct (single answer), incorrect, and answered (ambiguous) responses. Model - Llama-3.1-8b-Instruct.

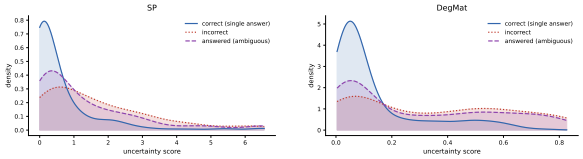



(a) Sequence Probability (b) Degree Matrix

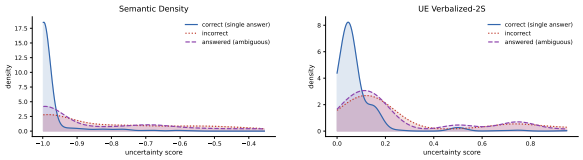



(c) Semantic Density (d) Verbalized-2S

Figure 10: Density of uncertainty scores for correct (single answer), correct (multi answer), and incorrect (pooled) responses. Model - Qwen-2.5-14b-Instruct.

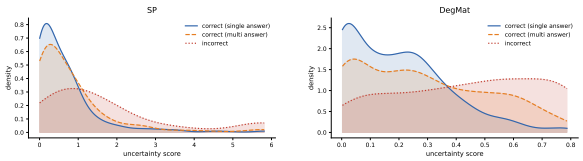



(a) Sequence Probability (b) Degree Matrix

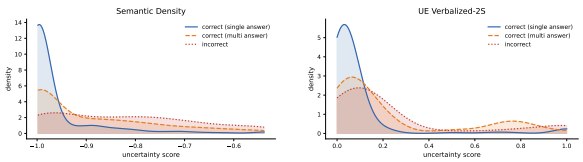



(c) Semantic Density (d) Verbalized-2S

Figure 11: Density of uncertainty scores for correct (single answer), incorrect, and answered (ambiguous) responses. Model - Qwen-2.5-14b-Instruct.

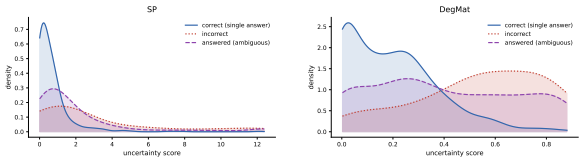



(a) Sequence Probability (b) Degree Matrix

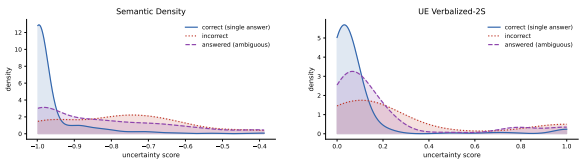



(c) Semantic Density (d) Verbalized-2S

Figure 12: Density of uncertainty scores for correct (single answer), correct (multi answer), and incorrect (pooled) responses. Model - Llama-3.3-70b-Instruct.

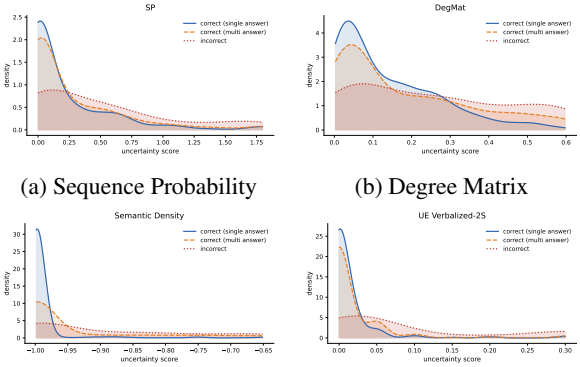



(a) Sequence Probability (b) Degree Matrix




(c) Semantic Density (d) Verbalized-2S

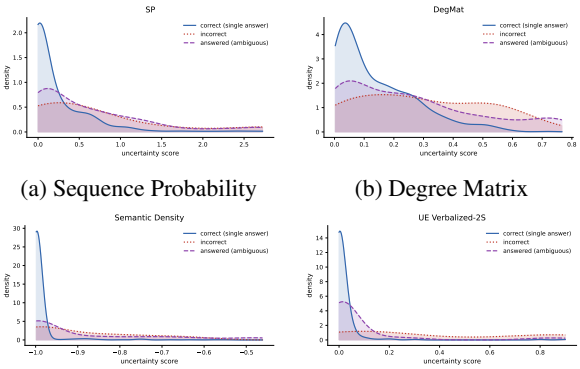
Figure 13: Density of uncertainty scores for correct (single answer), incorrect, and answered (ambiguous) responses. Model - Llama-3.3-70b-Instruct.




(a) Sequence Probability (b) Degree Matrix

(c) Semantic Density (d) Verbalized-2S

Figure 14: Density of uncertainty scores for correct (single answer), correct (multi answer), and incorrect (pooled) responses. Model - GPT-5.




(a) Sequence Probability (b) Degree Matrix

(c) Semantic Density (d) Verbalized-2S

Figure 15: Density of uncertainty scores for correct (single answer), incorrect, and answered (ambiguous) responses. Model - GPT-5.