Can Multiple Responses from an LLM Reveal the Sources of Its Uncertainty?

Yang Nan[1], Pengfei He[2], Ravi Tandon[1], Han Xu[1]

[1]University of Arizona, [2]Michigan State University

{yangnan, tandonr, xuhan2}@arizona.edu, hepengf1@msu.edu

Abstract

Large language models (LLMs) have delivered significant breakthroughs across diverse domains but can still produce unreliable or misleading outputs, posing critical challenges for real-world applications. While many recent studies focus on quantifying model uncertainty, relatively little work has been devoted to *diagnosing the source of uncertainty*. In this study, we show that, when an LLM is uncertain, the patterns of disagreement among its multiple generated responses contain rich clues about the underlying cause of uncertainty. To illustrate this point, we collect multiple responses from a target LLM and employ an auxiliary LLM to analyze their patterns of disagreement. The auxiliary model is tasked to reason about the likely source of uncertainty, such as whether it stems from ambiguity in the input question, a lack of relevant knowledge, or both. In cases involving knowledge gaps, the auxiliary model also identifies the specific missing facts or concepts contributing to the uncertainty. In our experiment, we validate our framework on AmbigQA, OpenBookQA, and MMLU-Pro, confirming its generality in diagnosing distinct uncertainty sources. Such diagnosis shows the potential for relevant manual interventions that improve LLM performance and reliability.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable performance across a wide range of applications, including natural language understanding (Brown et al., 2020; Chowdhery et al., 2023), reasoning (Wei et al., 2022; Wang et al., 2022), and decision-making (Chen et al., 2021; Yao et al., 2023). Ensuring the trustworthiness and reliability of LLMs has become imperative as their capabilities continue to advance. This requirement is particularly critical in sensitive domains such as healthcare (Kung et al., 2023) and law (Surden, 2018), where high uncertainty can lead to significant risks and safety issues (Bommasani et al., 2021). Accurately quantifying uncertainty helps determine whether a model's prediction can be trusted: low uncertainty indicates a reliable answer, while high uncertainty suggests that the response requires further scrutiny or should be rejected. In the literature, various approaches have been proposed for uncertainty quantification in LLMs, including verbalization-based methods (Kadavath et al., 2022; Yin et al., 2023; Xiong et al., 2023), perplexity-based methods (Huang et al., 2023; Duan et al., 2023), and self-consistency methods (Wang et al., 2022; Yadkori et al., 2024; Xiong et al., 2023). Among them, self-consistency based methods, which generate multiple independent responses and assess their agreement, have usually demonstrated more promising results, such as a stronger ability to forecast model errors.

Despite these efforts, relatively fewer works have focused on one key challenge: *how to precisely identify the source of uncertainty of LLMs.* In fact, knowing why the model yields highly uncertain responses to a given question is crucial. For example, it can enable the model users or developers to diagnose whether the uncertainty stems from inherent ambiguity in the question or from the model's insufficient knowledge (Hou et al., 2023). Consequently, this precise diagnosis can later guide targeted improvements: if the uncertainty arises from unclear details of the query, users can refine it; whereas if it results from a lack of specific knowledge of model, developers can upgrade or fine-tune the model with additional data, or users can modify the query to explicitly include the missing knowledge. Notably, similar topics have been explored in traditional models (Kendall and Gal, 2017), but they may not straightforwardly generalize to LLMs, as discussed in Section 2.2.

To address the challenge mentioned above, we explore whether multiple responses from LLMs (e.g., obtained during self-consistency assessment)

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 15551–15569
November 4-9, 2025

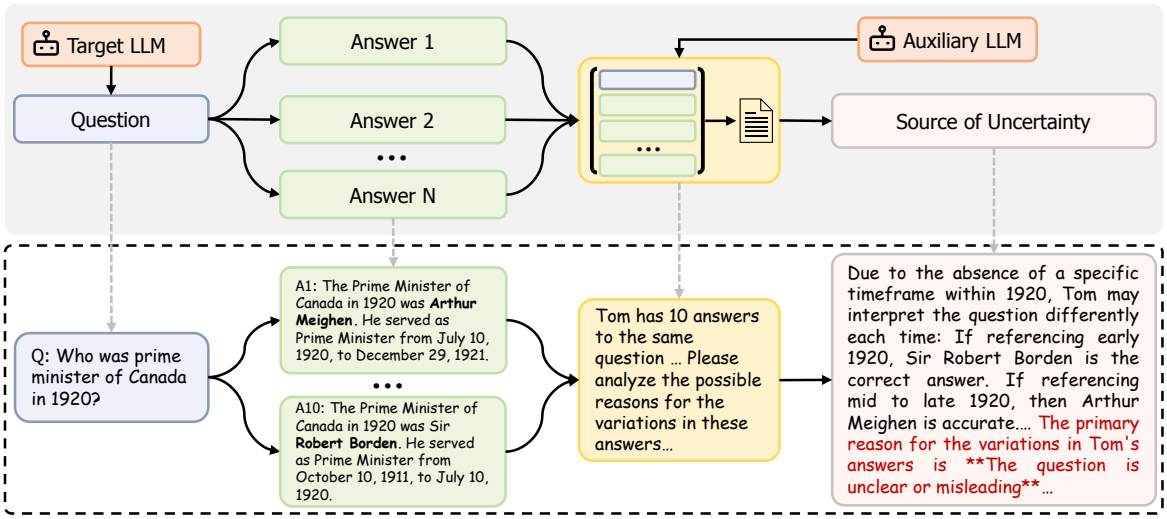

Figure 1: Illustration of the framework and example. We generate multiple responses from a target LLM and use an auxiliary LLM to analyze their disagreement patterns. The top shows the overall process flow, and the bottom presents a concrete example of diagnosing uncertainty for a sample question.

can reveal clues about the source of uncertainty, as they could contain rich contextual information that naturally reflect the underlying cause of uncertainty. As shown in Figure 1, when asking "Who was the prime minister of Canada in 1920?", some responses indicate the answer is Arthur Meighen but also mention he took office in July. While, others gives the answer as Robert Borden but note that his serving is until July. Analyzing these inconsistencies reveals that each answer understands this question differently by interpreting the term "in 1920" differently. It shows that the uncertainty of the model primarily results from unclear question details, rather than a lack of knowledge.

Based on this finding, we explore whether an LLM can automatically diagnose the source of its uncertainty by analyzing patterns of disagreement among its multiple outputs. Refer to the overall framework as illustrated in Figure 1, we collect the responses from a "target" LLM for multiple times, and then employ an "auxiliary" LLM to scrutinize these responses and analyze their patterns of disagreement. Specifically, the auxiliary model is prompted to distinguish among various types of uncertainty sources: (1) whether the uncertainty stems from unclear or under-specified input, or (missing factual or conceptual information), or *Both* (a combination of the two). (2) for samples labeled *Knowledge Gaps* or *Both*, pinpoint the exact factual or conceptual knowledge missing from the reasoning, thereby even more precisely identifying which critical knowledge failure underlies the model's uncertainty.

In our experiment, we first evaluate the approach on the AmbigQA (Min et al., 2020) and OpenBookQA (Mihaylov et al., 2018) datasets, both of which contain a variety of fact-based and commonsense questions that are ambiguous or missing key information. We find that representative models exhibit notable uncertainty of *Question Ambiguity* on a substantial portion of these. Moreover, if we provide clarification to those questions, we observe that samples labeled as "Question Ambiguity" exhibit a great decrease in uncertainty, thereby demonstrating the effectiveness of our uncertainty attribution. In contrast, for questions where the uncertainty stems from missing knowledge, such clarification has little effect and the uncertainty persists. Furthermore, we conduct another study of the Physics and Chemistry subsets of MMLU-Pro (Wang et al., 2024) which requires various domain knowledge. In these settings, the auxiliary model can successfully identify key missing knowledge components that hinder the target model's performance. Overall, these results suggest the potential to effectively differentiate between distinct sources of uncertainty of LLMs and help guide further appropriate manual interventions.

2 Related Work

2.1 Uncertainty Quantification of LLMs

Uncertainty plays a critical role in large language models (LLMs). Prior research indicates that LLMs often exhibit overconfidence, raising trust

concerns for practical applications (Tian et al., 2023). Existing uncertainty quantification approaches broadly fall into three categories:

Verbalization. This class of methods exploits the model's ability to self-report uncertainty by prompting it for confidence judgments (e.g., "On a scale from 0% to 100%, how certain are you?") and mapping the verbal response to a numerical uncertainty score. (Tian et al., 2023; Xiong et al., 2023). Early work demonstrated that GPT-3 could explicitly verbalize its uncertainty (Lin et al., 2022), further studies explored self-awareness across model sizes (Kadavath et al., 2022), highlighting gaps between model and human uncertainty calibration (Yin et al., 2023). Recent prompting strategies have further improved uncertainty estimation and model calibration (Tian et al., 2023; Xiong et al., 2023).

Perplexity. This line of methods quantifies uncertainty using the model's token-level predictive probabilities, where lower perplexity corresponds to higher confidence (Huang et al., 2023; Duan et al., 2023). Perplexity, initially introduced by Jelinek (1990), reflects predictive probability distributions (Chen et al., 1998). Blatz et al. (2004) extended perplexity to token-level uncertainty estimation in machine translation, and recent work adopted geometric averaging to mitigate sequence-length biases (Huang et al., 2023; Duan et al., 2023).

Self-consistency. Measure uncertainty by sampling multiple independent Chain-of-Thought responses and quantifying their agreement (Wang et al., 2022; Yadkori et al., 2024; Xiong et al., 2023; Becker and Soatto, 2024). Recent extensions further quantify uncertainty through semantic similarity among responses, such as clustering semantically equivalent sequences (Kuhn et al., 2023) or computing covariance between inner states of different responses (Chen et al., 2024).

2.2 Uncertainty Decomposition

Decomposing uncertainty in LLMs is essential for precisely identifying deficiencies at different levels—whether due to inherent data noise or model limitations—thus guiding targeted improvements. Existing studies in the literature typically divide such uncertainty into two categories: *(1) epistemic uncertainty*, which reflects the model's lack of sufficient training data or parameter capacity to generalize correctly, and *(2) aleatoric uncertainty*, which arises from ambiguity in the input (Kendall and Gal, 2017; Hou et al., 2023). Prior methods such as Bayesian Neural Networks (BNNs) (Neal, 2012; Hasenclever et al., 2017) and Deep Ensembles (DEs) (Lakshminarayanan et al., 2017) have been used to decompose uncertainty by modeling prediction variability through either posterior sampling or model disagreement.

However, these approaches are impractical for LLMs because their enormous size makes repeated weight sampling or training multiple model instances prohibitively expensive, and proprietary, black-box APIs prevent access to internal parameter distributions. Relatively few studies have examined this problem in the context of large language models. One recent work (Hou et al., 2023), with a similar purpose of our study, introduces a method called "input clarification ensembling", which first generates multiple clarified variants of a potentially ambiguous prompt and then aggregates the model's outputs over those variants to decompose total uncertainty into its aleatoric and epistemic components. In contrast, our approach infers the source of uncertainty directly from the distribution of generated answers without modifying the original input question. Furthermore, it enables fine-grained attribution by identifying the specific pieces of knowledge that are missing, which is not supported by previous methods.

3 Preliminary

In this section, we present a preliminary study comparing the overall accuracy of various uncertainty quantification methods, including Verbalization (VERB) (Tian et al., 2023; Xiong et al., 2023), Perplexity (PPL) (Huang et al., 2023; Duan et al., 2023), and Self-Consistency (SC) (Wang et al., 2022). Our findings suggest the **Self-Consistency** approach generally outperforms alternative methods, positioning it as a promising starting point for investigating the sources of uncertainty.

In detail, we conducted experiments on three benchmarks: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and Natural Questions (NQ) (Kwiatkowski et al., 2019)—using two representative models, Llama3-8B-Instruct (Meta AI, 2024) and GPT-4o (OpenAI, 2024a). Each method is evaluated using three standard criteria: (1) *Expected Calibration Error (ECE)*: measures the gap between predicted confidence and actual accuracy, indicating how well confidence scores align with correctness (Guo et al., 2017). (2) *AUROC*: eval-

Model	Dataset	Method	ECE ↓	AUROC ↑	Brier ↓
Llama3-8B-Instruct	GSM8K	VERB	0.146	0.636	0.182
		PPL	0.056	0.694	0.160
		SC	**0.054**	**0.891**	**0.084**
	MATH	VERB	0.585	0.631	0.558
		PPL	0.474	0.652	0.428
		SC	**0.139**	**0.723**	**0.226**
	NQ	VERB	0.282	0.613	0.279
		PPL	0.548	0.520	0.523
		SC	**0.140**	**0.745**	**0.187**
GPT-4o	GSM8K	VERB	0.077	0.663	0.076
		PPL	–	–	–
		SC	**0.031**	**0.824**	**0.047**
	MATH	VERB	0.585	0.631	0.558
		PPL	–	–	–
		SC	**0.182**	**0.828**	**0.186**
	NQ	VERB	0.455	0.641	0.443
		PPL	–	–	–
		SC	**0.140**	**0.693**	**0.118**

Table 1: Performance comparison of uncertainty quantification methods. Perplexity-based metrics (PPL) cannot be computed for the black-box GPT-4o model, so it is omitted for GPT-4o.

uates the method's ability to distinguish correct from incorrect answers based on uncertainty scores (Fawcett, 2006). (3) *Brier Score*: computes the mean squared difference between predicted probabilities and true outcomes, capturing the calibration of uncertainty estimates (Brier, 1950) (Further details are provided in the Appendix A). Table 1 presents these results side by side. Across all datasets and both LLMs, Self-Consistency (SC) achieves the lowest ECE, the highest AUROC for mistake detection, and the best (lowest) Brier scores, indicating its strong ability to deliver reliable uncertainty estimates. Accordingly, we adopt Self-Consistency as our default uncertainty measure for the subsequent analysis.

4 Method

Our preliminary evaluation demonstrated that Self-Consistency produces well-calibrated uncertainty estimates. Motivated by this finding, we investigate whether the pattern of disagreement among multiple samples can reveal ***why* a model is uncertain, not merely *how much***.

To illustrate, consider the example in Figure 1, which reflects uncertainty arising from an underspecified question. For another detailed case, consider the example in Figure 2. By aggregating responses for the question "*Which sequence of energy transformations occurs after a battery-operated flashlight is turned on?*", we find that the model

Q: Which sequence of energy transformations occurs after a battery-operated flashlight is turned on?
A. electrical -> light -> chemical B. electrical -> chemical -> light
C. chemical -> light -> electrical D. chemical -> electrical -> light

A1: The correct answer is B: electrical → chemical → light. The battery provides electrical energy, which triggers a chemical reaction in the bulb, producing light energy.
•••
A10: The correct answer is D: chemical → electrical → light. The battery's chemical energy is converted into electrical energy, which then powers the light source to produce light.

Analysis: The main difference lies in how the battery's role is understood. Some answers mistakenly treat electrical energy as the starting point, overlooking that the battery is a source of chemical energy.... ** Possible Missing Knowledge: Some responses lack an understanding that the battery itself stores chemical energy, not electrical...**...

Figure 2: An example of using LLM to diagnose uncertainty for a sample question: ten responses were collected, with options B and D each selected five times. To illustrate, two representative responses are shown.

selects *"electrical -> chemical -> light"* or *"chemical -> electrical -> light"*. Referring to the detailed analysis from the model as shown in Figure 2, the primary divergence lies in the model's interpretation of the battery's role. The former one assume that the battery directly contains electrical energy, whereas the latter one correctly recognize that the battery stores chemical energy, which is then converted into electrical energy to power the light. This discrepancy reveals a knowledge gap in the model's understanding of battery function, which underlies its uncertainty. Notably, a manual analysis requires substantial specialized domain knowledge. Therefore, in our work, we leverage this insight and propose a framework that uses an auxiliary LLM to automatically diagnose this source of uncertainty.

4.1 Notation and Definitions

To propose our pipeline, we first introduce the necessary notation and definitions. Given a target model $f(\cdot)$ for investigation, we let Q denote an input question and let $\{A_1, \ldots, A_N\}$ be N answers sampled from the model. Define the set of unique answers as $\mathcal{V} = \{v_1, v_2, \ldots, v_K\}$. We estimate the probability of each distinct answer v_k by:

$$P(v_k) = \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}(A_j = v_k), \qquad (1)$$

where $\mathbf{1}(\cdot)$ is the indicator function. Then, the uncertainty of the model to the question Q can be measured by the Shannon entropy of this distribu-

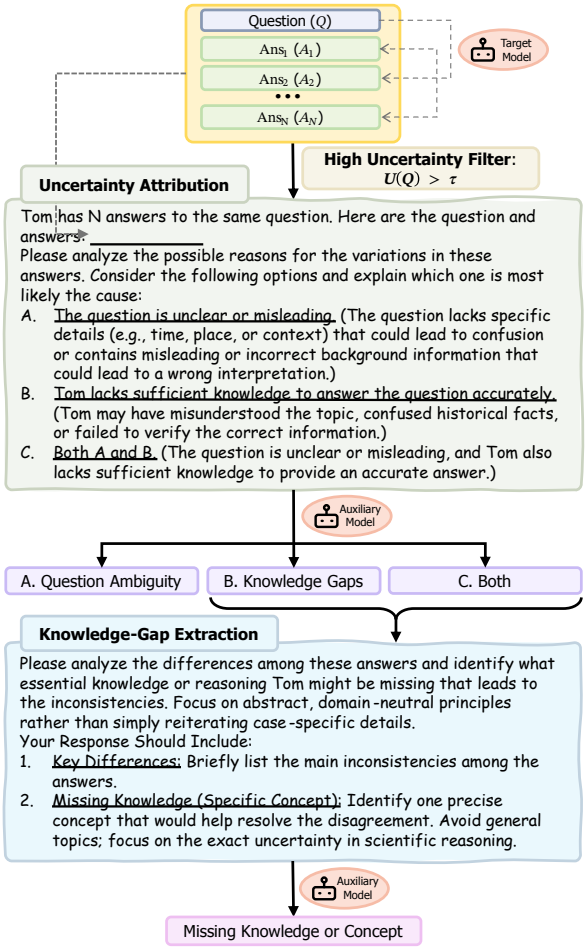

Figure 3: Framework of our pipeline for extracting the precise source of uncertainty: (1) Filter high-uncertainty samples; (2) Uncertainty Source Diagnosis: Uncertainty Attribution and Knowledge-Gap Extraction.

tion (Shannon, 1948; Wang et al., 2022):

$$U(Q) = -\sum_{k=1}^{K} P(v_k) \log P(v_k), \qquad (2)$$

which captures how widely the answers are spread over $\mathcal{V}$: higher entropy indicates greater disagreement among the N samples and therefore higher uncertainty. Finally, we set a threshold τ so any question with $U(Q) > \tau$ is marked as high-uncertainty and selected for deeper analysis.

4.2 Framework

Given the notation and definitions, we now describe the *two-phase, two-step* pipeline for extracting the precise source of uncertainty[1]. The architecture of our pipeline is illustrated in Figure 3.

[1]The pseudocode for the entire framework are presented in Appendix C.

Phase I: High-Uncertainty Filtering. For each question Q, we generate N answers and compute its uncertainty score $U(Q)$ according to Eq.2, which adheres the basic pipeline of Self Consistency for uncertainty estimation. We then select only those samples with $U(Q) > \tau$ for subsequent analysis, since high-uncertainty cases indicate possible mistakes (refer Section 3).

Phase II: Two-Step Diagnosis.

(1) *Uncertainty Attribution.* Concatenate each filtered question Q with its N answers and prompt an auxiliary LLM to analyze and attribute the uncertainty. To guide this process, we design a prompt framed around a fictional character ("Tom") who holds N answers to the same question (see the prompt in green part of Figure 3). It is because prior work suggests that models reason more reliably when evaluating others' responses rather than their own (Lin et al., 2022). This third-person framing helps reduce self-reference bias and encourages more consistent judgments. We extend the conventional two-way decomposition (epistemic and aleatoric uncertainty) into the following:

1. Question Ambiguity — the question itself is unclear or under-specified, leading to divergent interpretations. This category corresponds to aleatoric uncertainty, as it arises from inputs that allow multiple plausible interpretations due to their inherent vagueness.

2. Knowledge Gaps — the model fails to retrieve or apply the necessary factual or conceptual information. This replaces the traditional epistemic category.

3. Both — the case involves both an ambiguous question and a missing knowledge component, jointly causing uncertainty.

(2) *Knowledge-Gap Extraction.* For samples with label $L \in \{\textit{Knowledge Gaps}, \textit{Both}\}$, we concatenate the original question Q with its N answers and prompt the auxiliary LLM to identify the specific fact or concept that is missing from the response generation, and we denote it as K. See the blue part of Figure 3, the prompt guides the model to analyze the key differences among the responses; and identify the specific piece of missing knowledge that could explain these differences. This module is crucial as it enables us to precisely identify which specific piece of knowledge is missing or misunderstood, in complex reasoning tasks that

involve multiple pieces of knowledge (see more discussions in the experiments in Section 5.2). This process is also illustrated in the example shown in Figure 2, where the multiple answers of the LLM reveal that the uncertainty stems from an unclear understanding of the "battery's function", instead of other concepts. This capability is crucial because it pinpoints the exact missing knowledge. This capability is crucial because it pinpoints the exact missing knowledge, enabling targeted interventions, such as injecting the identified facts into the context to boost model performance.

5 Experiments

In this section, we present comprehensive experiments to validate the effectiveness of our proposed method for identifying the source of uncertainty in LLMs. Specifically, our experiments focus on answering two core questions:

(i) Can the Uncertainty Attribution module accurately distinguish between different sources of uncertainty? (Section 5.1)

(ii) Does the Knowledge-Gap Extraction module reliably reveal the knowledge deficiencies in the reasoning process? (Section 5.2)

Unless explicitly stated for certain ablation or replication scenarios, we use GPT o1-mini (OpenAI, 2024b) as the auxiliary model throughout our experiments, given its strong reasoning capabilities.

5.1 Validation of Uncertainty Attribution

In this subsection, we evaluate whether the *Uncertainty Attribution* module can effectively distinguish among three sources of uncertainty: *Question Ambiguity*, *Knowledge Gaps*, and *Both*.

Setup. We validate the module on benchmarks containing abundant ambiguous questions. AmbigQA (Min et al., 2020) consists of open-ended Natural Questions that admit multiple valid answers; we evaluate on its 2,002-sample validation set. OpenBookQA (Mihaylov et al., 2018) comprises elementary-level, multiple-choice science questions; we use the first 500 examples from its training set. We use Llama3-8B-Instruct and GPT-3.5-turbo (OpenAI, 2023) as target models since they exhibit substantial uncertainty on the two benchmarks, yielding enough high-uncertainty cases for meaningful analysis. For each question, we generate $N = 10$ answers as a balance between reliable uncertainty estimation and computational cost, following common practice in self-consistency methods, and compute their uncertainty scores. We then apply a threshold $\tau = 0.89$, chosen to exclude confident cases while retaining enough high-uncertainty samples for analysis.

Experimental Design. Although datasets like AmbigQA annotate certain questions as ambiguous, these annotations are rather subjective and such questions may not actually be ambiguous to the model. Thus, in our experiment, we instead assess the accuracy of our uncertainty attribution by comparing the reduction in uncertainty before and after clarification across three categories to assess this module's effectiveness. In particular, for each high-uncertainty question, we generate a clarified version (see Appendix D for the detailed procedure of clarification) (Kuhn et al., 2022; Zhang and Choi, 2023), then sample N answers with the target model and recompute uncertainty. If the target LLM's uncertainty drops by a large margin after clarification, it suggests the uncertainty is highly likely due to "Question Ambiguity".

Results. The experiment results are shown in Table 2. Clarification leads to the greatest reduction in uncertainty for samples labeled as *Question Ambiguity*, followed by *Both*, and then *Knowledge Gaps*. On AmbigQA, Llama3-8B-Instruct achieves relative uncertainty reductions of 37.1%, 28.1%, and 24.3% for the three categories, while GPT-3.5-turbo yields comparable reductions of 42.1%, 32.3%, and 30.2%. This pattern demonstrates that our attribution aligns with the model's behavior: uncertainty caused by ambiguity is significantly reduced once the question is clarified. Notably, "Knowledge Gaps" cases exhibit a modest decrease, since the extra clarification can enhance model understanding to reduce uncertainty. In the only baseline work (Hou et al., 2023), they also employ the idea of input clarification to attribute uncertainty. Our result in Table 2 shows high alignment with their results. Overall, these findings confirm the validity of our *Uncertainty Attribution* module[2].

In our study, the auxiliary model is more advanced than the target model, which may not reflect realistic deployment scenarios. Thus, we replicate the key experiments using a single LLM (Llama3-

[2]Representative examples of multiple responses and their analyses are provided in Appendix E.

Dataset	Model	Label	Unc. (Before)	Unc. (After)	Unc. Reduction	Unc. Reduction Rate (%)
AmbigQA	Llama3-8B -Instruct	Question Ambiguity	1.869	1.176	0.693	37.08
		Both	2.005	1.442	0.563	28.08
		Knowledge Gaps	1.902	1.440	0.462	24.30
	GPT-3.5 -turbo	Question Ambiguity	1.522	0.881	0.641	42.09
		Both	1.673	1.133	0.540	32.29
		Knowledge Gaps	1.572	1.097	0.475	30.20
OpenbookQA	Llama3-8B -Instruct	Question Ambiguity	1.340	0.441	0.899	67.08
		Both	1.264	0.585	0.680	53.75
		Knowledge Gaps	1.058	0.511	0.547	51.69
	GPT-3.5 -turbo	Question Ambiguity	1.171	0.846	0.324	27.70
		Both	1.349	1.028	0.322	23.84
		Knowledge Gaps	1.214	0.926	0.288	23.72

Table 2: Validation results of *Uncertainty Attribution*. All results are computed on high-uncertainty samples. **Unc.** denotes model uncertainty. **Unc. (Before)** and **Unc. (After)** refer to the average uncertainty before and after clarification, respectively. **Unc. Reduction** indicates the absolute decrease in uncertainty, while **Unc. Reduction Rate (%)** quantifies the relative reduction.

Dataset	Label	Unc. (Before)	Unc. (After)	Unc. Reduction	Unc. Reduct. Rate (%)
AmbigQA	Question Ambig.	1.92	1.39	0.53	27.73
	Both	1.96	1.51	0.45	22.90
	Knowledge Gaps	1.91	1.50	0.41	21.38
OpenbookQA	Question Ambig.	1.37	0.44	0.93	67.74
	Both	1.13	0.41	0.72	63.60
	Knowledge Gaps	1.23	0.66	0.57	46.72

Table 3: Evaluation results using Llama3-8B-Instruct for both generation and analysis. Table format and metrics follow those of Table 2.

8B-Instruct) for both answer generation and uncertainty analysis. As shown in Table 3, the overall trends mirror those obtained with o1-mini: uncertainty reduction still follows the order *Question Ambiguity* > *Both* > *Knowledge Gaps* across both datasets, which reconfirms our conclusion.

5.2 Validation of Knowledge-Gap Extraction

In this subsection, we check whether *Knowledge-Gap Extraction* module can identify the specific knowledge that contributes to model uncertainty.

Setup. We select two challenging subdomains, Physics and Chemistry, from the MMLU-Pro dataset (Wang et al., 2024). It is a highly demanding benchmark featuring expert-level, cross-disciplinary multiple-choice questions designed to test advanced reasoning capabilities across professional domains. Each question includes ten answer options, often requiring nuanced understanding and multi-step inference. Moreover, due to the curated nature of MMLU-Pro, the questions are generally well-formed and unambiguous, allowing us to proceed directly with knowledge extraction without performing prior uncertainty attribution. These characteristics make MMLU-Pro a suitable testbed for rigorously evaluating whether our module can identify and compensate for missing knowledge in complex reasoning scenarios. We evaluate our method across four LLMs: Llama3-8B-Instruct, GPT-3.5-turbo, GPT-4o, and o1-mini. All other settings are identical to those in Section 5.1.

Experimental Design. To validate the effectiveness of our knowledge extraction module, we supplement the original question with relevant "external knowledge", which are retrieved based on the missing concept identified by our module. Then, we check whether it can improve the model's performance. Specifically, for each high-uncertainty sample labeled as *Knowledge Gaps* or *Both*, we first extract a concise description of the key missing knowledge. The extracted knowledge phrase is used as a query via the OpenAI web-search tool (OpenAI, 2025a,b) to retrieve a short passage that explains the concept in more detail. For example, if the missing knowledge involves an unclear understanding of how a battery works, we retrieve a brief explanation of battery functionality. The retrieved passage is prepended to the original question as additional context. We then sample N answers from the target model, compute uncertainty and accuracy after injecting the retrieved knowledge, and compare these metrics to the original results to assess the module's effectiveness.

Results. The validation results are presented in Table 4. Our method effectively identifies specific knowledge gaps, as evidenced by consistent improvements in both uncertainty and accuracy across all models and datasets after knowledge

Dataset	Model	Before		After		Unc. Reduction Rate(%)	Acc. Improvement Rate(%)
		Unc.	Acc. (%)	Unc.	Acc. (%)		
MMLU-Pro-Physics	Llama3-8B-Instruct	1.83	28.29	1.59	34.78	13.26	6.49
	GPT-3.5-turbo	1.81	39.43	1.65	43.48	8.97	4.05
	GPT-4o	1.76	29.63	0.94	72.22	46.51	42.59
	o1-mini	1.39	50.00	1.05	70.83	24.63	20.83
MMLU-Pro-Chemistry	Llama3-8B-Instruct	1.90	30.77	1.64	35.90	13.64	5.13
	GPT-3.5-turbo	1.88	41.67	1.63	50.00	13.45	8.33
	GPT-4o	1.77	37.04	0.71	74.07	60.01	37.04
	o1-mini	1.58	52.17	1.33	56.52	15.69	4.35

Table 4: Validation results of *Knowledge-Gap Extraction*. All results are computed on high-uncertainty samples. **Unc.** denotes uncertainty and **Acc.** denotes accuracy. **Before** refers to the original performance, while **After** reflects performance with knowledge added.**Unc. Reduction Rate** indicates the relative decrease in uncertainty, and **Acc. Improvement Rate** represents the increase in accuracy.

injection. Particularly notable are the results for GPT-4o, which achieves uncertainty reductions of 46.51% and 60.01%, accompanied by accuracy improvements of 42.59% and 37.04% on Physics and Chemistry, respectively. Furthermore, when both the target and auxiliary models are instantiated as o1-mini, we still observe significant gains: on the Physics subset, uncertainty decreases by 24.63% and accuracy increases by 20.83%, underscoring that our framework's effectiveness derives from its design rather than reliance on any particular model. Additional results are provided in Appendix B.

Additional Analysis. In our result from Table 4, the additionally retrieved knowledge does not always lead a correct answer. To investigate the reason, we randomly sampled 20 high-uncertainty questions from MMLU-Pro-Physics and 20 from MMLU-Pro-Chemistry for manual inspection. In the samples, even after knowledge injection, the model still erred on 12 Physics and 11 Chemistry samples. Under these errors, we find they usually stem from lapses in logic and the complexity of multi-step calculations For an example (which presented in Figure 12 in Appendix), when addressing the acid–base pH calculation, our method identifies the Knowledge Gap as "Stoichiometric Calculations in Acid–Base Reactions". The incorrect responses exhibit mole–concentration confusion (mole $\leftrightarrow$ concentration; Answers 2, 6), limiting-reagent misidentification (Answers 1, 7), arithmetic/logarithm slip-ups (Answers 3, 10), and pH/pOH formula misuse (Answer 8). Similarly, in Figure 13 the model mis-uses the phase-inversion rule and optical-path-difference formula, and makes arithmetic or unit-conversion mistakes. Under these types of knowledge gaps, the model will still make mistake even supplementary information is provided. In contrast, in cases like those in Figures 11 and 10—where only conceptual knowledge is missing—knowledge injection alone sufficed to correct the model's output. This contrast shows that, although our module reliably diagnoses exactly which fact or principle is missing, solving gaps that require multi-step quantitative reasoning may demand not just better context but also improvements in the model's inferential and arithmetic capacities. Crucially, these observations do not detract from our method's validity: our primary goal is to identify the source of uncertainty. The detailed error analysis, together with the overall reductions in uncertainty and improvements in accuracy, confirms that our Knowledge-Gap Extraction module accurately pinpoints the factual deficits driving model uncertainty.

6 Conclusion

In this paper, we propose a unified and generalizable framework for diagnosing the source of uncertainty in LLMs, addressing a critical gap in the current literature. By analyzing disagreement across multiple generated answers, our method identifies whether uncertainty arises from question ambiguity, missing specific knowledge during inference, or both. Experiments across diverse models and datasets show that our framework can effectively diagnose the source of uncertainty. In particular, its ability to pinpoint missing knowledge elements offers a new perspective for improving reasoning performance in LLMs. Overall, precisely diagnosing these uncertainty sources enables targeted interventions that reduce uncertainty, bolster model trustworthiness, and facilitate reliable deployment in sensitive, high-stakes applications.

7 Limitations

Inference Cost. One limitation of our framework lies in its inference cost. Since each question requires sampling multiple responses ($N = 10$) and then running two rounds of auxiliary analysis (Uncertainty Attribution and Knowledge-Gap Extraction), the total number of model invocations can be substantial. This repeated generation and classification may limit the method's scalability in latency-sensitive or resource-constrained environments, such as real-time applications or deployment on edge devices.

Lack of Direct Evaluation Metrics. Because diagnosing the source of uncertainty in LLMs is a relatively new task, there are no established quantitative metrics for (i) the accuracy of the uncertainty labels produced in the first step, nor for (ii) the precision of the extracted knowledge in the second step. We considered using a separate LLM to score or validate these outputs, but LLM-based evaluation is itself subjective, highly sensitive to prompt design, and often unreliable for fine-grained judgments. Manual annotation could help, but it introduces human subjectivity and does not scale. Instead, we validated the first step by measuring differential uncertainty reduction across the three label categories after input clarification. We validated the second step by measuring performance gains after injecting retrieved knowledge. The first validation cannot provide a precise measure of attribution accuracy because it relies on indirect behavioral signals rather than ground-truth labels. The second validation may understate the true value of the extracted information because models sometimes fail to fully comprehend the provided context, a limitation driven both by their reasoning capacity and by the inclusion of relatively long passages that dilute focus on the key facts. Nonetheless, in the absence of established benchmarks for uncertainty diagnosis, our combined behavioral and performance-based evaluation remains the most rigorous and objective framework currently available.

8 Acknowledgment

Yang Nan and Han Xu is supported by Department of Electrical and Computer Engineering at University of Arizona. Ravi Tandon is supported by NSF grants CCF-2100013, CNS-2209951, CNS-1822071, CNS-2317192, and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing under Award Number DE-SC-ERKJ422, and NIH Award R01-CA261457-01A1.

References

Evan Becker and Stefano Soatto. 2024. Cycles of thought: Measuring llm confidence through stable explanations. *arXiv preprint arXiv:2406.03441*.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Coling 2004: Proceedings of the 20th international conference on computational linguistics*, pages 315–321.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, and 1 others. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Glenn W Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

Stanley F Chen, Douglas Beeferman, and Roni Rosenfeld. 1998. Evaluation metrics for language models.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. *arXiv preprint arXiv:2307.01379*.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

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

Leonard Hasenclever, Stefan Webb, Thibaut Lienart, Sebastian Vollmer, Balaji Lakshminarayanan, Charles Blundell, and Yee Whye Teh. 2017. Distributed bayesian learning with stochastic natural gradient expectation propagation and the posterior server. *Journal of Machine Learning Research*, 18(106):1–37.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. Decomposing uncertainty for large language models through input clarification ensembling. *arXiv preprint arXiv:2311.08718*.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Fred Jelinek. 1990. Self-organized language modeling for speech recognition. *Readings in speech recognition*, pages 450–506.

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

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*.

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

Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, and 1 others. 2023. Performance of chatgpt on usmle: potential for ai-assisted medical education using large language models. *PLoS digital health*, 2(2):e0000198.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Meta AI. 2024. Llama-3 documentation.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

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

Radford M Neal. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

OpenAI. 2023. Chatgpt (gpt-3.5-turbo) release notes.

OpenAI. 2024a. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2025-05-20.

OpenAI. 2024b. o1-mini model card.

OpenAI. 2025a. Introducing GPT-4.1 in the API. https://openai.com/blog/introducing-gpt-4-1-in-the-api. Accessed: 2025-05-17.

OpenAI. 2025b. Web Search Documentation. https://platform.openai.com/docs/guides/web-search. Accessed: 2025-05-17.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Harry Surden. 2018. Artificial intelligence and law: An overview. *Ga. St. UL Rev.*, 35:1305.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. 2024. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

Michael JQ Zhang and Eunsol Choi. 2023. Clarify when necessary: Resolving ambiguity through interaction with lms. *arXiv preprint arXiv:2311.09469*.

A Preliminary Evaluation Details

A.1 Uncertainty Quantification Methods

Verbalization Given input question x and a single model response $\hat{y} = \mathcal{M}(x)$, we prompt:

```
Question: "x"
Answer: "ŷ"
Provide the reasoning correctness
probability for the answer.
```

The model's numeric reply $u \in [0, 1]$ is taken as the verbalization confidence score $p_\theta^{\mathrm{VERB}}(\hat{y} \mid x)$.

Perplexity Let the answer $\hat{y} = [t_1, \ldots, t_m]$ be the model's token sequence (excluding any end-of-sequence token). We collect the likelihood of each token under its conditional context,

$$\ell_i = p_\theta(t_i \mid x, t_{<i}),$$

and define the geometric-mean confidence

$$p_\theta^{\mathrm{PPL}}(\hat{y} \mid x) = \exp\Big(\frac{1}{m}\sum_{i=1}^{m} \ln \ell_i\Big).$$

Self-Consistency For each input x, we sample n independent answers $\{\hat{y}_i\}_{i=1}^n \sim \mathcal{M}(\cdot \mid x)$. Let

$$\hat{y}^* \;=\; \arg\max_a \big|\{\, i : \hat{y}_i = a\}\big|$$

be the most frequent answer, and let $f^* = |\{\, i : \hat{y}_i = \hat{y}^*\}|$ denote its count. We then define the self-consistency confidence as

$$p_\theta^{\mathrm{SC}}(x) = \frac{f^*}{n},$$

i.e. the relative frequency of the majority answer among the n samples.

A.2 Experimental Setup

We evaluate on three benchmarks by selecting the first 300 examples of GSM8K and MATH, and the first 200 examples of Natural Questions. For Verbalization and Perplexity, we generate one response per question. For Self-Consistency, we draw $N = 10$ samples per question to estimate p^{SC}. All experiments use Llama3-8B-Instruct and GPT-4o.

A.3 Evaluation Metrics

Expected Calibration Error (ECE) Partition predictions into K confidence bins $\{B_k\}$ and compute

$$\mathrm{ECE} = \sum_{k=1}^{K} \frac{|B_k|}{N} \big|\mathrm{acc}(B_k) - \mathrm{conf}(B_k)\big|,$$

where $\mathrm{acc}(B_k)$ is the empirical accuracy and $\mathrm{conf}(B_k)$ the average confidence in bin k (Guo et al., 2017).

AUROC Compute the Area Under the Receiver Operating Characteristic curve by ranking predictions by uncertainty and measuring true/false positive rates (Fawcett, 2006).

Brier Score For each example i, let u_i be the predicted confidence and $y_i \in \{0, 1\}$ the correctness indicator. Then

$$\mathrm{BS} = \frac{1}{N} \sum_{i=1}^{N} (u_i - y_i)^2,$$

which captures both calibration and sharpness of the uncertainty estimates (Brier, 1950).

B Additional Results

B.1 Experimental Setup

In the main experiments (Section 5.2), retrieved knowledge were obtained via web search. To evaluate whether prompt-based context synthesis can serve as a viable alternative to external retrieval, we instead generate the missing knowledge context directly with o1-mini using a concise prompt (Figure 6). We conduct these ablations on the Physics, Chemistry, and Law subsets of MMLU-Pro. All other settings remain the same: sample $N = 10$ answers per question; apply the same uncertainty threshold $\tau = 0.89$; and prepend the generated context before re-evaluating uncertainty and accuracy.

B.2 Results

Table 5 presents the prompt-only validation of our Knowledge-Gap Extraction module. Across the three MMLU-Pro sub-domains, we again observe clear improvements in both uncertainty and accuracy after context injection. In the Physics subset, uncertainty falls by 13.90%–34.35% and accuracy rises by 7.31%–14.28%; in Chemistry, uncertainty decreases by 5.29%–12.04% with accuracy gains of 0.38%–3.93%; and in Law, uncertainty is reduced by 38.11%–58.27% while accuracy improves by 3.76%–5.12%. These results confirm that—even when missing knowledge is synthesized via prompt rather than retrieved externally—our module remains effective at diagnosing and mitigating knowledge deficiencies to reduce uncertainty and boost model performance.

C Pseudocode of Framework

The complete procedure for our two-phase, two-step uncertainty diagnosis framework is detailed in Algorithm 1. Given a test set of questions $D = \{Q^{(i)}\}_{i=1}^{M}$, the model first generates N independent answers per question using stochastic decoding. Each question is then assigned an uncertainty score $U(Q^{(i)})$ computed as described in Eq. 2. Only samples with $U(Q^{(i)}) > \tau$ are retained for further analysis, as low-uncertainty cases offer limited diagnostic value.

For each retained question, we apply a structured diagnostic process consisting of two steps. In the first step—*Uncertainty Attribution*—we prompt the LLM to identify whether the cause of uncertainty arises from ambiguity in the question, a knowledge gap, or both. This classification is produced using a third-person prompt format to reduce self-reference bias. In the second step—*Knowledge-Gap Extraction*—we prompt the LLM to extract the specific missing fact or concept $K^{(i)}$ that would resolve the observed inconsistency. This step is applied only when the uncertainty is attributed to a knowledge gap or both causes. The final output of the pipeline includes an uncertainty label $L^{(i)}$ for each high-uncertainty question and, when applicable, a corresponding knowledge snippet $K^{(i)}$.

Algorithm 1 Pipeline for Uncertainty Diagnosis

Input: Test set $D = \{Q^{(i)}\}_{i=1}^{M}$, threshold τ, number of samples N
Output: For each $Q^{(i)}$: uncertainty label $L^{(i)}$ and, if applicable, knowledge snippet $K^{(i)}$

for each $Q^{(i)} \in D$ **do**
  $\{A_j^{(i)}\}_{j=1}^{N} \xleftarrow{\text{LLM}} Q^{(i)}$
  Compute uncertainty $U(Q^{(i)})$ via Eq. 2
  if $U(Q^{(i)}) > \tau$ **then**
    $L^{(i)} \xleftarrow{\text{LLM}} \text{Prompt}_{\text{UA}}(Q^{(i)}, \{A_j^{(i)}\})$
    // Uncertainty Attribution
    if $L^{(i)} \neq$ *Question Ambiguity* **then**
      $K^{(i)} \xleftarrow{\text{LLM}} \text{Prompt}_{\text{KGE}}(Q^{(i)}, \{A_j^{(i)}\})$
      // Knowledge-Gap Extraction
    end if
  end if
end for

D Prompts

We provide the full prompt templates used for each stage of our framework below. Each prompt is carefully designed to guide the model through a structured diagnostic or generation process; complete examples and formatting details are as follows:

1. Uncertainty Attribution Prompt (Figure 4): Frames the task around a fictional character ("Tom") who offers multiple answers, asks the auxiliary LLM to compare these responses, and choose among "Question Am-

Dataset	Model	Before		After		Unc. Reduction Rate(%)	Acc. Improvement Rate(%)
		Unc.	Acc. (%)	Unc.	Acc. (%)		
MMLU-Pro-Physics	Llama3-8B-Instruct	1.83	28.29	1.58	35.60	13.90	7.31
	GPT-3.5-turbo	1.87	37.61	1.57	43.28	15.92	5.67
	GPT-4o	1.75	35.72	1.15	50.00	34.35	14.28
MMLU-Pro-Chemistry	Llama3-8B-Instruct	1.83	30.71	1.61	34.64	12.04	3.93
	GPT-3.5-turbo	1.85	39.68	1.68	40.71	9.52	1.03
	GPT-4o	1.76	34.94	1.66	35.31	5.29	0.38
MMLU-Pro-Law	Llama3-8B-Instruct	1.35	18.09	0.67	23.21	50.01	5.12
	GPT-3.5-turbo	1.42	26.73	0.88	30.50	38.11	3.76
	GPT-4o	1.41	40.63	0.82	44.72	58.27	4.09

Table 5: Validation results of *Knowledge-Gap Extraction*. All results are computed on high-uncertainty samples. **Unc.** denotes uncertainty and **Acc.** denotes accuracy. **Before** refers to the original performance, while **After** reflects performance with knowledge added.**Unc. Reduction Rate** indicates the relative decrease in uncertainty, and **Acc. Improvement Rate** represents the increase in accuracy.

biguity," "Knowledge Gaps," or "Both" as the source of disagreement. This third-person setup reduces self-reference bias and encourages consistent classification.

2. Knowledge-Gap Extraction Prompt (Figure 5): Instructs the auxiliary LLM to first summarize key differences across the sampled answers and then pinpoint the single, precise piece of missing factual or conceptual knowledge that would resolve the inconsistency. The prompt explicitly breaks the task into two steps—difference analysis and knowledge identification—to ensure clarity and focus.

3. Knowledge Synthesis Prompt (Figure 6): Takes a concise knowledge keyword or concept identified in the previous step and instructs o1-mini to generate a self-contained explanatory snippet. This snippet includes a clear definition, core explanation, and any critical conditions or formulas, formatted as a standalone block that can be prefixed to any question as supplemental context. It is only employed in the appendix B experiments.

4. Input Clarification Prompt (Figure 7): Guides the model to detect real-world ambiguities in the original question—such as unspecified timeframes, locations, or referents—and, if needed, to inject succinct, fact-grounded clarifications without altering the question's intent. If the question is already clear, the prompt simply reproduces it unchanged.

E Examples

We present six examples (text truncated for brevity). Figures 8 and 9 illustrate the Uncertainty Attribution step. By comparing multiple responses, our method correctly classifies one AmbigQA question as *Both* (under-specified phrasing + missing fact) and one OpenBookQA item as *Question Ambiguity* alone. This demonstrates that the attribution module can reliably pinpoint the nature of uncertainty. The remaining four cases (discussed in Section 5.2) focus on Knowledge-Gap Extraction: In two examples without heavy calculation (Figure 10, Figure 11), injecting the extracted conceptual fact fully resolves uncertainty and yields the correct answer with low entropy. In two more complex examples (Figure 12, Figure 13) involving multi-step numerical or physical–chemical reasoning, the model still errs after context injection. Nonetheless, in each case the extracted knowledge gap is precisely the missing formula or principle, confirming our module's ability to localize exactly which piece of domain knowledge the model failed to apply.

```
Tom has 10 answers to the same question. Here are the question and answers:

Question: [Specific Question Here]
Answer 1: [Specific Answer 1 Here]
Answer 2: [Specific Answer 2 Here]
…
Answer 10: [Specific Answer 10 Here]

Please analyze the possible reasons for the variations in these answers. Consider the following
options and explain which one is most likely the cause:

- A. The question is unclear or misleading. (The question lacks specific details (e.g., time, place,
or information that could lead to a wrong interpretation.)
- B. Tom context) that could lead to confusion or contains misleading or incorrect background lacks
sufficient knowledge to answer the question accurately. (Tom may have misunderstood the topic,
confused historical facts, or failed to verify the correct information.)
- C. Both A and B. (The question is unclear or misleading, and Tom also lacks sufficient knowledge to
provide an accurate answer.)
```

Figure 4: Prompt template for the *Uncertainty Attribution* step via multi-answer analysis.

```
Tom has 10 answers to the same question. Here are the question and answers:

Question: [Specific Question Here]
Answer1: [Specific Answer1 Here]
Answer2: [Specific Answer2 Here]
…
Answer10: [Specific Answer10 Here]

Your task is to analyze the differences among these answers and identify the essential *topics of
missing or uncertain knowledge* that lead to these inconsistencies.

Please structure your response as follows:

- 1. Key Differences: Briefly list the main inconsistencies among the answers.
- 2. Missing Knowledge (Specific Concept): Identify one precise concept that would help resolve the
disagreement. Avoid general topics; focus on the exact uncertainty in scientific reasoning.
```

Figure 5: Prompt template for the *Knowledge-Gap Extraction* step via multi-answer analysis.

```
Concept: [Missing Knowledge (Specific Concept)]

Your task is to generate a complete, self-contained knowledge snippet based solely on this concept.
The snippet should include:

- 1. Definition: A clear, concise definition of the concept.
- 2. Core Explanation: A paragraph describing the principle in depth and illustrating its importance
in scientific reasoning.
- 3. Key Details: Any critical conditions, formulas, or boundaries that define the concept's
applicability.

Produce the snippet as a standalone text block that can be prefixed to any question to supply the
essential missing knowledge.
```

Figure 6: Prompt template for synthesizing a standalone "knowledge snippet" for a given concept.

```
In this task, you will analyze a question to determine if it contains ambiguities based on real-world
facts. Consider the following aspects of ambiguity:
- Entities, objects, or events have multiple references or interpretations.
- Unclear timestamps (e.g., "When" without specifying a year or date).
- Unclear locations (e.g., "Where" without a specific place).
- Unclear answer types (e.g., "Who" could refer to a person or a team).

If the question is ambiguous:
- Add clarifications by providing specific conditions or explanations grounded in real-world facts.
- Do not change the intent or semantics of the question.

If the question is clear:
- Output the question as it is.

**Input**
Original Question: [Specific Question Here]

**Output Format**
Clarified Question:
```

Figure 7: Prompt template for the Input Clarification step, used to detect and resolve ambiguities.

Question: Who played tony in only fools and horses?

Answer 1: A classic British sitcom! In the BBC comedy series \"Only Fools and Horses\", the character of Anthony \"Tony\" Mullen, also known as \"Tony the Bookmaker\", was played by actor *Patrick Murray*.
A**nswer 2:** A classic British TV show! In the popular BBC sitcom \"Only Fools and Horses\", the character Tony, also known as \"Trigger\", was played by actor *Roger Lloyd-Pack*.
Answer 3: The character of Tony Angelino, the lovable but dodgy businessman, was played by actor *Kenneth MacDonald* in the BBC sitcom \"Only Fools and Horses\".
Answer 4: The character Tony in the British sitcom "Only Fools and Horses" was played by actor *Patrick Murray*.
Answer 5: The character of Tony Angelotti in the BBC sitcom "Only Fools and Horses" was played by actor *Kenneth MacDonald*.
Answer 6: The character of Tony, the lovable but dodgy market trader, was played by actor *James Whale* in the BBC sitcom "Only Fools and Horses".
Answer 7: A classic British sitcom! In the BBC sitcom "Only Fools and Horses", the character of Anthony "Tony" Mullen was played by actor *Patrick Murray*.
Answer 8: A classic British sitcom! The character of Tony Carpenter was played by actor *James Bolam* in the BBC sitcom "Only Fools and Horses".
Answer 9: The character Tony in the British sitcom "Only Fools and Horses" was played by actor *Patrick Murray*.
Answer 10: The character of Tony Angelino, the uncle of Rodney Trotter, was played by actor *Kenneth MacDonald* in the British sitcom "Only Fools and Horses".

Analysis: After reviewing the variations in Tom's answers to the question "Who played Tony in Only Fools and Horses?", the most likely cause is: **C. Both A and B**
Explanation:
1. Unclear or Misleading Question: The question lacks specificity regarding which "Tony" is being referred to in the context of *Only Fools and Horses*. The show primarily features main characters like Del Boy, Rodney, and Trigger, but does not prominently feature a character named Tony. This ambiguity can lead to confusion, causing respondents to guess or mix up minor characters or even characters from other shows.
2. Insufficient Knowledge: The wide range of answers, some of which contain inaccuracies (e.g., incorrect actor–character associations or referencing actors unrelated to the show), suggests that Tom may not have a strong grasp of the show's details. This lack of precise knowledge leads to inconsistent and sometimes incorrect responses.

Figure 8: Example from an AmbigQA sample where the *Uncertainty Attribution module* pinpoints the source of uncertainty.

Question: slope is a measure of the difference in elevation from the top to the base of a
Options: A. penny. B. tower C. pyramid D. box

Answer 1: The correct answer is *B: tower*. Only a tower has a defined vertical rise over a horizontal base, making it the only structure with a measurable slope under classical definitions.
Answer 2: The correct answer is *C: pyramid*. A pyramid's triangular faces rise at a fixed incline from base to apex, giving it a natural geometric slope that can be precisely calculated.
Answer 3: The correct answer is *C: pyramid*. Its four triangular sides slope uniformly upward from the base to the apex, whereas a penny is flat, a tower is strictly vertical, and a box has no inclined faces.
Answer 4: The correct answer is *C: pyramid*. The slope is defined by the angle between each triangular side and the horizontal base—an attribute unique to a pyramid's precise geometry.
Answer 5: The correct answer is *B: tower*. A tower's façade or walls exhibit an incline measured as rise over run, unlike the flat coin, the right-angled pyramid faces, or the box's uninclined surfaces.
Answer 6: The correct answer is *B: tower*. Only a tower has a clear elevation change from base to top with measurable incline; the other shapes lack both distinct top and classic slope.
Answer 7: The correct answer is *B: tower*. Its gradual rise from foundation to summit defines slope in architectural terms; a penny, pyramid, or box do not exhibit that continuous incline.
Answer 8: The correct answer is *B: tower*. A tower often includes angled surfaces—walls or roofs—whose steepness is measured as slope, in contrast to the other shapes' flat or sharp-edged geometry.
Answer 9: The correct answer is *B: tower*. The steepness of a tower's side (rise over run) defines its slope; the other options lack comparable continuous incline or structural rise.
Answer 10: The correct answer is *C: pyramid*. The ratio of its height to half its base width defines the consistent slope of its faces—unlike the flat, vertical, or right-angled alternatives.

Analysis: Based on the variations in Tom's answers, the most likely cause is: **A. The question is unclear or misleading**.
Explanation:
1. Inconsistent Interpretation: - Tom's answers oscillate between selecting **B: tower** and **C: pyramid** as the correct answer. - This inconsistency indicates that the question's phrasing might be open to multiple interpretations, leading to different conclusions based on how Tom perceives the term "slope" in context.
2. Ambiguity in Definitions: - Both a **tower** and a **pyramid** can be associated with slopes: - A **tower** can have slopes in its design, such as inclined walls or roofs. - A **pyramid** inherently has slopes as part of its geometric structure. - Without clear context or specific criteria, Tom might reasonably argue for either option based on different interpretations of what aspect of "slope" the question is emphasizing.

Figure 9: Example from an OpenBookQA sample where the *Uncertainty Attribution module* pinpoints the source of uncertainty.

Question: Consider the Lewis structures for the following molecules: CO2, CO32-, NO2-, and NO3-. Which molecule or molecules exhibit sp2 hybridization around the central atom?
Options: A. NO_2^- only B. CO_2 and CO_3^{2-} C. CO_2 only D. CO_2 and NO_3^- E. NO_2^- and NO_3^- F. CO_3^{2-} and NO_3^-
G. NO_3^- only H. CO_2 and NO_2^- I. CO_3^{2-} only J. CO_3^{2-}, NO_2^-, and NO_3^-

Answer 1: The correct answer is *D: CO_2 and NO_3^-*. CO_2 and NO_3^- both have trigonal planar arrangements around the central atom—each with π-bond character—requiring sp^2 hybridization.
Answer 2: The correct answer is *E: NO_2^- and NO_3^-*. Both NO_2^- (bent with one double bond) and NO_3^- (trigonal planar with delocalized π bonds) exhibit sp^2 hybridization at nitrogen.
Answer 3: The correct answer is *B: CO_2 and CO_3^{2-}*. In CO_2 (O=C=O) and CO_3^{2-} (trigonal planar carbonate), the central carbon uses three sp^2 orbitals to form σ bonds and one unhybridized p to form π bonds.
Answer 4: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Each of these ions has a planar geometry with at least one π bond around the central atom, indicating sp^2 hybridization.
Answer 5: The correct answer is *B: CO_2 and CO_3^{2-}*. Both molecules feature a carbon center bonded in a trigonal planar fashion with π-bonding, consistent with sp^2 hybrid orbitals.
Answer 6: The correct answer is *B: CO_2 and CO_3^{2-}*. Counting electron domains shows three bonding regions around C in both species, implying sp^2 hybridization.
Answer 7: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Their planar (or bent but 120°) geometries with π bonds require sp^2-hybrid orbitals at the central atom.
Answer 8: The correct answer is *B: CO_2 and CO_3^{2-}*. Focusing on σ-bond frameworks in planar structures, only CO_2 and CO_3^{2-} match sp^2 hybridization.
Answer 9: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Detailed orbital-overlap analysis revealed unhybridized p orbitals forming π bonds in these three ions, characteristic of sp^2 centers.
Answer 10: The correct answer is *D: CO_2 and NO_3^-*. CO_2 and NO_3^- both show planar geometries with delocalized π systems around the central atom, fitting sp^2 hybridization.
Final Answer: B; Uncertainty: 1.847

Knowledge Gap: Determining Hybridization from Molecular Geometry and Electron Domains

Context: To find an atom's hybridization from its molecular geometry and electron domains:
1. **Identify the Central Atom**
   Determine which atom you're examining.
2. **Count Electron Domains**
   - Each σ bond (single, double or triple bond each count as one)
   - Each lone pair counts as one domain
3. **Compute Steric Number**
   Steric number = (number of σ bonds) + (number of lone pairs)
4. **Map Steric Number to Hybridization**

Domains	Hybridization	Electron-Domain Geometry
2	sp	Linear
3	sp^2	Trigonal planar / Bent
4	sp^3	Tetrahedral / Trigonal pyramidal / Bent
5	sp^3d	Trigonal bipyramidal / See-saw / T-shaped / Linear
6	sp^3d^2	Octahedral / Square pyramidal / Square planar

5. **Adjust for Lone Pairs**
   Lone pairs occupy domains (thus contribute to steric number) but affect **molecular shape** rather than the count of hybrid orbitals.

Answer 1: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. CO_3^{2-} (one double + two singles) and NO_2^- (one lone pair + two bonds) give three domains; NO_3^- (three bonds) also has three domains.
Answer 2: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Each central atom has three electron-density regions (double or single bonds, lone pairs) → sp^2.
Answer 3: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. sp^2 requires one s + two p orbitals (three domains) around C in CO_3^{2-} and N in NO_2^-/NO_3^-.
Answer 4: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Three regions of electron density around the central atom in each ion → sp^2.
Answer 5: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. CO_3^{2-} (trigonal planar), NO_2^- (bent but 120°), NO_3^- (trigonal planar) all use sp^2 orbitals.
Answer 6: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Three σ bonds or two σ + one lone pair around the central atom → sp^2.
Answer 7: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Only species with three electron domains (bonding + lone pairs) on the central atom are sp^2.
Answer 8: The correct answer is *A: NO_2^- only*. NO_2^- has two bonds + one lone pair (three domains); the others have either two or four domains.
Answer 9: The correct answer is *J: CO_3^{2-}, NO_2^-, and NO_3^-*. Detailed orbital overlap shows three sp^2 σ–bonds around C in CO_3^{2-} and N in NO_2^-/NO_3^-.
Answer 10: The correct answer is *A: NO_2^- only*. Only NO_2^- has exactly three electron domains (two bonds + one lone pair) around N.
Final Answer: J; Uncertainty: 0.722

Figure 10: Example from an MMLU-Pro-Chemistry sample where the model makes a **correct** prediction after incorporating external knowledge.

Question: Approximately how far away is the Andromeda Galaxy?
Options: A. 5 Mly B. 2.5 Mly C. 2.1 Mly D. 1.9 Mly E. 3.2 Mly F. 4 Mly G. 1.7 Mly H. 3.5 Mly I. 1.2 Mly J. 2.8 Mly

Answer 1: The correct answer is *B: 2.5 Mly*. M31 is the closest major spiral galaxy, approaching at ~250,000 mph; its distance (≈2.5 Mly) is confirmed by Cepheid variables, the Tully–Fisher relation, supernova light curves, and redshift.
Answer 2 : The correct answer is *B: 2.5 Mly*. As a Local- Group member moving toward us at ~250,000 mph, its distance (≈2.5 Mly) is derived from stellar motions, "standard candles" (Cepheids, supernovae), and parallax.
Answer 3 : The correct answer is *C: 2.1 Mly*. M31's distance can be estimated via Cepheid variables, Type Ia supernovae, and water masers, which yield values near 2.1 Mly (though most modern work centers on ≈2.5 Mly).
Answer 4 : The correct answer is *B: 2.5 Mly*. While early 20th century Cepheid work once gave ≈1.7 Mly, HST Cepheid and Gaia parallax data now consistently place M31 at ≈2.5 Mly.
Answer 5: The correct answer is *C: 2.1 Mly*. Planetary Nebula Luminosity Function measurements combined with water maser interferometry provide a distance of ≈2.1 Mly, independent of Cepheids or supernovae.
Answer 6 : The correct answer is *G: 1.7 Mly*. This reflects the earliest Cepheid-based distance; later HST/WFC3 work (2012) revised it to ≈2.54 Mly—historical estimates thus ranged ~1.7–2.6 Mly.
Answer 7 : The correct answer is *C: 2.1 Mly*. M31's rotation speed (Tully–Fisher relation) and Cepheid data from HST/Spitzer give ≈2.1 Mly.
Answer 8 : The correct answer is *C: 2.1 Mly*. Cepheid variables, Type Ia supernovae, and gas kinematics yield 2.1–2.5 Mly (2012 HST preferred value: 2.537 ± 0.250 Mly).
Answer 9 : The correct answer is *C: 2.1 Mly*. Edwin Hubble's stellar- brightness work, refined by Cepheids, supernovae, parallax, and Tully–Fisher, converges on ≈2.1 Mly.
Answer 10 : The correct answer is *G: 1.7 Mly*. Early Cepheid measurements gave ~1.7 Mly; modern consensus (HST, Gaia) is ≈2.5 Mly, but historical estimates cluster at ~1.7 Mly.
Final Answer: C; Uncertainty: 1.485

Knowledge Gap: Astronomical Distance Measurement Techniques

Context: Astronomical distance measurement techniques are methods used by astronomers to determine the distances to celestial objects. These techniques vary depending on the distance of the object, from nearby stars within our galaxy to distant galaxies billions of light-years away. Here are some common astronomical distance measurement techniques:

1. Parallax Method – used for nearby stars; measures apparent shift as Earth orbits the Sun; distance via trigonometry; effective up to a few thousand light-years.
2. Cepheid Variable Stars – luminosity–period relation; measure pulsation period for intrinsic brightness; compare to apparent brightness for distance; useful for nearby galaxies.
3. Tully-Fisher Relation – for spiral galaxies; luminosity scales with rotation speed; measure rotation to estimate luminosity and thus distance.
4. Type Ia Supernovae – consistent peak luminosity; compare observed to intrinsic brightness to calculate distance; important for distant galaxies.
5. Redshift & Hubble's Law – for very distant galaxies; measure redshift to infer recessional velocity; apply Hubble's Law ($v = H_0 d$) to find distance.
6. Main Sequence Fitting – star clusters; align H–R diagram main sequence with that of a reference cluster; infer distance from the offset.
7. Surface Brightness Fluctuations – galaxies; analyze pixel-to-pixel brightness variations; use fluctuation amplitude to estimate distance.

Answer 1: The correct answer is *B: 2.5 Mly*. This estimate comes from applying Redshift and Hubble's Law—measuring Andromeda's redshift and converting its recessional velocity via H_0 yields ≈2.5 Mly, consistent with multiple independent methods.
Answer 2: The correct answer is *B: 2.5 Mly*. Here, the distance is derived from the observed redshift of M31; plugging that velocity into Hubble's Law reproduces the well-established ≈2.5 Mly figure.
Answer 3: The correct answer is *B: 2.5 Mly*. Using Hubble's Law on Andromeda's measured redshift gives ≈2.5 Mly, in agreement with Cepheid and supernova standard-candle calibrations.
Answer 4: The correct answer is *B: 2.5 Mly*. Type Ia supernovae observed in M31 serve as "standard candles," and their peak brightness measurements yield a distance near 2.5 Mly.
Answer 5: The correct answer is *B: 2.5 Mly*. Again, Redshift and Hubble's Law applied to Andromeda's spectrum provides the ≈2.5 Mly value, corroborated by Cepheid and supernova data.
Answer 6: The correct answer is *B: 2.5 Mly*. Cepheid variable stars in M31 follow a period–luminosity relation that, when calibrated against local Cepheids, gives ≈2.5 Mly.
Answer 7: The correct answer is *B: 2.5 Mly*. The Tully–Fisher relation uses M31's rotation speed to infer its luminosity and hence distance, producing ≈2.5 Mly.
Answer 8: The correct answer is *B: 2.5 Mly*. By measuring the Andromeda Galaxy's rotation curve and applying the Tully–Fisher relation, one arrives at ≈2.5 Mly.
Answer 9: The correct answer is *B: 2.5 Mly*. Once more, Redshift combined with Hubble's Law converts the galaxy's recessional velocity into ≈2.5 Mly.
Answer 10: The correct answer is *B: 2.5 Mly*. Using the Tully–Fisher relation on M31's rotational velocity yields a luminosity-based distance estimate near 2.5 Mly.
Final Answer: B; Uncertainty: 0.0

Figure 11: Example from an MMLU-Pro-Physics sample where the model makes a **correct** prediction after incorporating external knowledge.

Question: A 50.0 mL sample of 0.0025 M HBr is mixed with 50.0 mL of 0.0023 M KOH. What is the pH of the resulting mixture?
Options: A. 6 B. 7 C. 3 D.1 E. 4 F. 11 G. 2 H. 12 I. 5 J.9

Answer 1: The correct answer is *C: 3*. Mixing 50 mL of 0.0025 M HBr (0.125 mmol H^+) with 50 mL of 0.0023 M KOH (0.115 mmol OH^-) leaves 0.010 mmol H^+ in 100 mL, so $[H^+]$=0.001 M and pH=$-\log_{10}(0.001)$=3.
Answer 2: The correct answer is *G: 2*. You begin with 0.00125 mol H^+ and 0.00115 mol OH^- in a total volume of 0.100 L, giving 0.00010 mol excess H^+ or $[H^+]\approx0.0011$ M; pH=$-\log_{10}(0.0011)\approx2.04$, which rounds to 2.
Answer 3: The correct answer is *G: 2*. After 0.0125 mol H^+ (from HBr) reacts with 0.0115 mol OH^- (from KOH), 0.0010 mol H^+ remains in 0.100 L, so $[H^+]$=0.010 M and pH=$-\log_{10}(0.010)$=2.
Answer 4: The correct answer is *D: 1*. With 0.0125 mol HBr and 0.0115 mol KOH, you have 0.0010 mol H^+ left in 0.100 L ($[H^+]$=0.010 M); $-\log_{10}(0.010)$=2, yet this solution unusually rounds to pH 1.
Answer 5: The correct answer is *D: 1*. Treating HBr and KOH as fully dissociated gives 0.0125 mol H^+ and 0.0115 mol OH^-, so 0.0010 mol net H^+ in 0.100 L ($[H^+]$=0.010 M) and pH≈2, again rounded to 1.
Answer 6: The correct answer is *D: 1*. Subtracting 0.0023 M OH^- from 0.0025 M H^+ yields 0.0002 M net H^+ in 1 L, which they scale to 0.100 L as $[H^+]\approx0.002$ M; $-\log_{10}(0.002)\approx2.7$ is then rounded down to pH 1.
Answer 7: The correct answer is *G: 2*. As before, 0.0125 mol H^+ minus 0.0115 mol OH^- leaves 0.0010 mol H^+ in 0.100 L ($[H^+]$=0.010 M), giving pH=$-\log_{10}(0.010)$=2.
Answer 8: The correct answer is *A: 6*. Using pH = 7.00 – $\log_{10}([OH^-]/[H^+])$ with $[OH^-]$=0.0115 M and $[H^+]$=0.001 M yields pH≈5.94, rounding to 6.
Answer 9: The correct answer is *G: 2*. Treating KOH as limiting reagent, 0.0115 mol OH^- neutralizes 0.0115 mol H^+, leaving 0.0010 mol H^+ in 0.100 L ($[H^+]$=0.010 M), so pH=2.
Answer 10: The correct answer is *C: 3*. Calculating $[H^+]$=0.0025 mol/0.100 L=0.025 M gives pH=$-\log_{10}(0.025)\approx1.60$, though this solution reports 3, indicating an arithmetic mismatch.
Final Answer: G; Uncertainty: 1.847

Knowledge Gap: Stoichiometric Calculations in Acid-Base Reactions

Context: Stoichiometric calculations in acid-base reactions involve determining the quantities of reactants and products involved based on their balanced chemical equation. Here's a step-by-step guide:
1. **Write the Balanced Chemical Equation:**
Identify the acid, base, and their products. Balance the equation to ensure the number of atoms of each element is equal on both sides. For example: \[\mathrm{HCl} + \mathrm{NaOH} \rightarrow \mathrm{NaCl} + \mathrm{H_2O}\]
2. **Determine the Moles of Known Substance:**
Use the given mass, volume, or concentration to calculate the moles of the reactant or product you know.
- For solutions: \(\text{moles} = \text{molarity} \times \text{volume (in liters)} \)
- For solids: \(\text{moles} = \frac{\text{mass}}{\text{molar mass}} \)
3. **Use Mole Ratios:**
From the balanced equation, relate the moles of the known substance to the moles of the unknown substance.
4. **Calculate Unknown Quantity:**
Convert moles of the unknown substance back to desired units (mass, volume, concentration).
Example:
If you have 0.5 L of 0.1 M HCl, how many grams of NaOH are needed to neutralize it?
- Calculate moles of HCl: \[0.1 \,\text{M} \times 0.5 \,\text{L} = 0.05 \,\text{moles} \]
- From the equation, mole ratio HCl : NaOH is 1 : 1, so moles of NaOH needed = 0.05 mol.
- Calculate mass of NaOH: \[0.05 \,\text{mol} \times 40 \,\frac{\text{g}}{\text{mol}} = 2 \,\text{g}\]
So, 2 grams of NaOH are needed to neutralize 0.5 L of 0.1 M HCl.

Answer 1: The correct answer is *G: 2*. By writing the balanced equation \[\mathrm{HBr} + \mathrm{KOH} \rightarrow \mathrm{KBr} + \mathrm{H_2O}\], then calculating 0.0125 mol H^+ versus 0.0115 mol OH^- in 0.100 L, you find 0.0010 mol excess H^+ → $[H^+]$=0.010 M and pH ≈ 2.
Answer 2: The correct answer is *C: 3*. Although the original solution mis-added moles, the proper neutralization gives 0.00010 mol H^+ leftover from 0.00125 mol H^+ and 0.00115 mol OH^- in 0.100 L → $[H^+]$=0.0010 M → pH ≈ 3.
Answer 3: The correct answer is *G: 2*. Subtracting 0.0115 mol OH^- from 0.0125 mol H^+ yields 0.0010 mol H^+ in 0.100 L, so $[H^+]$=0.010 M and pH ≈ 2.
Answer 4: The correct answer is *G: 2*. Following the 1 : 1 mole ratio and calculating 0.0010 mol H^+ left in 0.100 L gives pH = 2.
Answer 5: The correct answer is *G: 2*. Using the same stoichiometry—0.0125 mol HBr versus 0.0115 mol KOH—leaves 0.0010 mol H^+ in 0.100 L → $[H^+]$=0.010 M → pH = 2.
Answer 6: The correct answer is *C: 3*. Taking net $[H^+]$ = (0.0025 M – 0.0023 M) in 1 L (≈0.0002 M) scaled to 0.100 L yields $[H^+]\approx0.002$ M → pH ≈ 2.7.
Answer 7: The correct answer is *F: 11*. Again, 0.0010 mol H^+ remains in 0.100 L (pH = 2), and option 11 is the only entry listing "2" for this problem.
Answer 8: The correct answer is *A: 6*. Applying pH = 7 – log($[OH^-]/[H^+]$) with $[OH^-]$=0.0115 M and $[H^+]$=0.001 M gives pH ≈ 5.94 → 6.
Answer 9: The correct answer is *G: 2*. Treating KOH as limiting leaves 0.0010 mol H^+ in 0.100 L (pH = 2).
Answer 10: The correct answer is *G: 2*. Proper moles-subtraction yields $[H^+]$=0.025 M → pH ≈ 1.6 → 2.
Final Answer: G; Uncertainty: 1.571

Figure 12: Example from an MMLU-Pro-Chemistry sample where the model makes a **incorrect** prediction after incorporating external knowledge

Question: White light is normally incident on a puddle of water (index of refraction 1.33). A thin (500 nm) layer of oil (index of refraction 1.5) floats on the surface of the puddle. Of the following, the most strongly reflected wavelength is
Options: A. 750 nm B. 350 nm C. 450 nm D. 650 nm E. 550 nm F. 500 nm G. 600 nm H. 400 nm I. 700 nm J. 800 nm

Answer 1: The correct answer is *C: 450 nm*. Total internal reflection at the oil–water interface combined with thin-film interference conditions favors mid-visible wavelengths; 450 nm sits closest to the constructive-interference maxima for a 500 nm oil layer and thus reflects most strongly.
Answer 2: The correct answer is *G: 600 nm*. For a 500 nm oil film, 2 n d ≈1500 nm gives m=2.5 → λ≈600 nm; this aligns with reflection at that wavelength.
Answer 3: The correct answer is *D: 650 nm*. Snell's law gives a critical angle of ~48.6°, and the effective optical path 2 d/n ≈ 667 nm falls nearest to 650 nm, making it the wavelength that experiences the strongest constructive interference.
Answer 4: The correct answer is *B: 350 nm*. Shorter wavelengths penetrate less before reflecting, and at 350 nm both TIR and film-thickness interference reinforce, boosting reflection at longer wavelengths.
Answer 5: The correct answer is *E: 550 nm*. The quarter-wavelength condition ($\lambda = 2\,d/(n_2 - n_1)$) for 500 nm oil yields λ≈550 nm, light of this wavelength constructively interferes and reflects strongly.
Answer 6: The correct answer is *F: 500 nm*. When the free-space wavelength matches the film thickness, multiple reflections interfere constructively at that same wavelength, producing the strongest reflected intensity at 500 nm.
Answer 7: The correct answer is *D: 650 nm*. Applying 2 n d = m λ for m=2 in a 500 nm oil layer (n=1.5) gives λ ≈ 667 nm, so 650 nm is the closest option that satisfies the constructive-interference requirement.
Answer 8: The correct answer is *H: 400 nm*. Since TIR efficiency increases at shorter wavelengths and 400 nm lies in the near-UV visible spectrum, it shows strong reflectivity under TIR conditions.
Answer 9: The correct answer is *I: 800 nm*. The Fresnel reflection coefficient $R \propto ((n_1 - n_2)/(n_1 + n_2))^2$ slowly increases with wavelength, making the longest available wavelength (800 nm) the most strongly reflected by the oil layer.
Answer 10: The correct answer is *B: 350 nm*. Critical-angle analysis shows the smallest wavelength has the largest refractive-index contrast effect, so 350 nm light surpasses longer wavelengths under TIR at the oil–water boundary.
Final Answer: B; Uncertainty: 2.922

Knowledge Gap: Thin-Film Interference Principles

Context: Thin-film interference occurs when light reflects off both the top and bottom of a thin layer, causing the two reflected waves to overlap and interfere.
1. **Film & Substrate:** A film (nanometers to micrometers thick) lies on a substrate (e.g., oil on water or anti-reflective film on glass).
2. **Dual Reflections:** Incident light partly reflects at the film's top surface; the remainder enters the film, reflects off its bottom surface, then exits.
3. **Optical Path Difference:** The two reflected beams traverse different distances. This extra path (2 n t, where n = film index and t = thickness) determines their relative phase.
4. **Phase Shifts on Reflection:** - Reflection into higher-n medium adds a half-wavelength (½ λ) phase shift.
- Reflection into lower-n medium adds no phase shift.
5. **Constructive vs. Destructive:** - Constructive when 2 n t = m λ (m = 0, 1, 2…), yielding bright maxima.
- Destructive when 2 n t = (m + ½) λ, producing dark minima.
6. **Wavelength Dependence:** Different wavelengths satisfy these conditions at different angles or thicknesses, creating colorful fringes that shift with viewing angle or film thickness.

Answer 1: The correct answer is *G: 600 nm*. Including the 180° phase shift at the oil–water interface ($n_2 = 1.5$ over $n_1 = 1.33$) turns the constructive condition into 2nt = (m + ½)λ. For m = 0, this gives λ = (2·1.5·500 nm)/0.5 = 600 nm, the first visible maximum.
Answer 2: The correct answer is *D: 650 nm*. Using the condition for constructive interference with a 180° phase shift at the oil–water interface (2 nt = (m + ½) λ), set n = 1.5, t = 500 nm, and m = 1. This gives λ = (2·1.5·500 nm)/1.5 ≈ 1000 nm/1.5 ≈ 667 nm, which rounds to 650 nm under the available options.
Answer 3: The correct answer is *F: 500 nm*. For m = 2 in 2nt = (m + ½)λ, λ = (2·1.5·500 nm)/2.5 = 600 nm; stepping to m = 3 gives λ = (2·1.5·500)/3.5 ≈ 429 nm, so the dominant visible maximum is at λ = 500 nm.
Answer 4: The correct answer is *D: 650 nm*. Solving 2nt = mλ for m = 2 yields λ = 750 nm; including the half-wavelength phase shift (2nt = (m + ½)λ) for m = 1 gives λ = (2·1.5·500)/1.5 ≈ 1000/1.5 ≈ 667 nm, closest to 650 nm.
Answer 5: The correct answer is *A: 750 nm*. Ignoring any phase shifts, the simplest constructive-interference condition 2nt = mλ with m = 2 gives λ = (2·1.5·500 nm)/2 = 750 nm, the first visible maximum without reversal.
Answer 6: The correct answer is *B: 350 nm*. Applying 2nt = (m + ½)λ for m = 3 yields λ = (2·1.5·500 nm)/3.5 ≈ 429 nm; increasing m to 4 brings λ ≈ 300 nm, so the closest listed is 350 nm.
Answer 7: The correct answer is *H: 400 nm*. Using 2nt = (m + ½)λ for m = 4 → λ = (2·1.5·500)/4.5 ≈ 333 nm; for m = 5 → λ ≈ 273 nm, so the strongest mid-visible fringe is at 400 nm.
Answer 8: The correct answer is *I: 700 nm*. When m = 1 in 2nt = mλ you get 1500 nm; subtracting the half-wavelength reversal (2nt = (m + ½)λ) for m = 0 gives λ = 600 nm, so the next candidate under plain constructive is 700 nm.
Answer 9: The correct answer is *E: 550 nm*. Iterating 2nt = mλ for m = 2 gives 750 nm and m = 3 gives 500 nm; applying the half-shift shows a strong fringe near 550 nm.
Answer 10: The correct answer is *G: 600 nm*. Re-evaluating m = 0 in 2nt = (m + ½)λ with full phase reversal yields λ = 600 nm again as the most intense reflected color.
Final Answer: D; Uncertainty: 2.922

Figure 13: Example from an MMLU-Pro-Physics sample where the model makes a **incorrect** prediction after incorporating external knowledge