Fine-grained Confidence Estimation for Spurious Correctness Detection in Large Language Models

Anonymous ACL submission

Abstract

In the deployment of Large Language Models (LLMs), "spurious correctness"—where answers are correct but reasoning contains errors—poses a critical risk by creating an illusion of reliability. While prior work on LLM confidence estimation focuses on answer-level or entire reasoning path confidence, these coarse-grained approaches fail to identify which specific parts of the reasoning contain errors. We propose a fine-grained confidence estimation framework that computes confidence scores for individual evidence triplets within reasoning chains, enabling precise localization of errors. We use special prompts to generates answers, evidence in triplet format, and their respective confidence scores simultaneously, allowing automatic detection of spurious correctness patterns where partial evidence contains factual errors. Evaluated on Japanese multihop QA across three model families representing different architectures and training approaches, we show that our approach exhibits superior calibration performance for evidence confidence and delivers strong ability to detect spurious correct answers (up to 84% discrimination accuracy). As a secondary benefit, joint generation of confidence scores improves answer confidence calibration by up to 43%. This prompt-based approach requires no model retraining and is immediately applicable to existing LLMs.

1 Introduction

As Large Language Models (LLMs) become increasingly deployed in real-world applications, the challenge of factuality—where LLMs generate information contradicting facts—remains one of the most critical issues (Huang et al., 2025; Min et al., 2023). One promising solution to this problem is confidence estimation, which aims to quantify the model's certainty in its outputs (Liu et al., 2025). Various approaches have been proposed to elicit *well-calibrated* confidence that aligns closely with

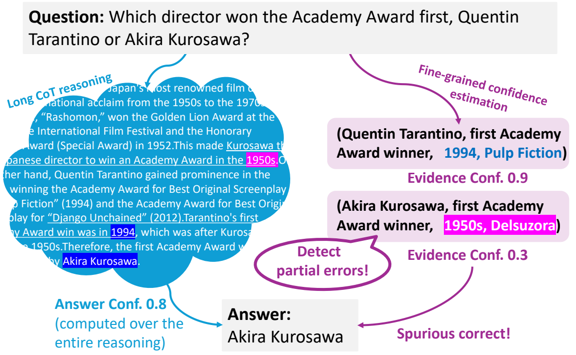
Question: Which director won the Academy Award first, Quentin Tarantino or Akira Kurosawa?
Long CoT reasoning
Japan's most renowned film d
national acclaim from the 1950s to the 1970
, "Rashomon," won the Golden Lion Award at the
e International Film Festival and the Honorary
Award (Special Award) in 1952.This made Kurosawa th
panese director to win an Academy Award in the 1950s.O
her hand, Quentin Tarantino gained prominence in the
winning the Academy Award for Best Original Screenplay
Fiction" (1994) and the Academy Award for Best Origi
play for "Django Unchained" (2012).Tarantino's first
y Award win was in 1994, which was after Kurosa
1950s.Therefore, the first Academy Award w
by Akira Kurosawa.
Fine-grained confidence estimation
(Quentin Tarantino, first Academy Award winner, 1994, Pulp Fiction)
Evidence Conf. 0.9
(Akira Kurosawa, first Academy Award winner, 1950s, Delsuzora)
Evidence Conf. 0.3
Detect partial errors!
Answer Conf. 0.8 (computed over the entire reasoning)
Answer: Akira Kurosawa
Spurious correct!

Figure 1: Overview of the proposed method: While long CoT reasoning makes error localization difficult and coarse-grained confidence masks specific mistakes, our fine-grained triplet-based confidence scores enable precise identification of incorrect components (e.g., the year Kurosawa won his first academy award, with the confidence of 0.3) within otherwise correct reasoning chains.

the correctness of the model's outputs. These approaches range from token probability-based methods (Kadavath et al., 2022), verbalized confidence (Tian et al., 2023) to consistency-based methods (Manakul et al., 2023).

A significant limitation of existing methods is that they estimate confidence at the level of *entire output*. In practice, however, responses from LLMs often consist of various components, including not just final answers but also intermediate reasoning steps, such as those produced in Chain-of-Thought (CoT) prompting (Wei et al., 2022). Consequently, assessing the confidence of *each component* separately allows LLMs to be more trustworthy, enabling users to better localize and interpret potential errors in LLM responses.

To address this limitation, we study methods for eliciting well-calibrated confidence for both intermediate reasoning steps and final answers from LLMs. As shown in Fig. 1, given a question (e.g., *Which director won...*), our method produces semi-structured evidence triplet as intermediate steps (e.g., (*Tarantino*, *first academy award winner*, *1994*)) along with real-valued confidence scores (e.g., 0.9) and then outputs the final answer. Following prior work, we adopt token-probability and prompt-based methods (Tian et al., 2023) for fine-grained confidence estimation over individual triplets.

To show the practical utility of fine-grained confidence, we apply it to the task of detecting *spuriously correct* answers, cases in which the final answers are correct but supported by incorrect evidence. This issue is particularly prominent in multihop QA task (Ishii et al., 2024a), with prior work observing it in 31% of instances in the JEMHopQA dataset (Ishii et al., 2024b).

Our main contributions are as follows:

1. We present the first study on fine-grained confidence estimation. Through a comprehensive analysis of five confidence extraction methods across three LLMs, we find that sampling-based methods yield better-calibrated confidence than other methods.

2. We demonstrate that fine-grained confidence scores better identify spuriously correct answers compared to conventional whole-output confidence scores, achieving an ROC-AUC of 0.84.

2 Related Work

2.1 LLM Confidence Estimation

LLM confidence estimation methods can be broadly categorized into three approaches:

Token probability-based methods: Kadavath et al. (2022) proposed estimating uncertainty directly from generation probabilities, though probability distributions are reported to be distorted in models trained with human preference optimization (Tian et al., 2023).

Linguistic confidence expression: Tian et al. (2023) demonstrated that for models trained with human preference optimization, prompting the model to self-report confidence—either as explicit numerical probabilities or as qualitative phrases such as "almost certain" or "likely"—produces better calibrated scores than relying on token probabilities alone.

Consistency-based methods: Manakul et al. (2023) proposed estimating confidence from agreement across multiple generation results. While computationally expensive, this enables more reliable estimation.

Importantly, none of these methods provide confidence scores at a granular level that would enable identification of specific erroneous components within reasoning chains. Our work addresses this gap by introducing fine-grained confidence estimation at the evidence triplet level.

2.2 Using Reasoning Process for Confidence Estimation

While several approaches leverage reasoning processes to improve answer confidence, they operate at coarse granularities:

Self-Consistency: Wang et al. (2022) samples multiple CoT reasoning paths and selects the most frequent answer. While each reasoning path can be considered an evidence, it does not score the correctness or reliability of the individual evidence.

Cycles of Thought: Becker and Soatto (2024) generates "answer + explanation" multiple times and quantifies uncertainty from explanation set stability. Their method uses explanation implication probabilities for weighting, but does not output confidence scores for the explanations themselves.

Confidence-based Self-Consistency: Taubenfeld et al. (2025) adds numerical confidence to the end of each reasoning path and selects final answers through weighted sums of identical answers. However, confidence evaluation of individual evidence elements is out of scope in this work.

These methods demonstrate the value of reasoning in confidence estimation but lack the critical granularity needed to pinpoint specific errors within reasoning chains. Our work extends these approaches by decomposing reasoning into evidence triplets and assigning confidence to each component independently.

2.3 The Spurious Correctness Problem

In multihop QA, the problem of "spurious correctness"—correct answers with incorrect reasoning—is severe. Prior research reports such cases amount to 31% of total instances (Ishii et al., 2024a).

However, these studies rely on manual evaluation, and to our knowledge no method targets automatic detection of spurious correctness in multihop QA using confidence scores.[1] In this work, we

[1] General hallucination detectors such as SelfCheckGPT (Manakul et al., 2023) focus on sentence-level factuality and do not distinguish correct answers with incorrect

enable automatic assessment of evidence/answer correctness and their confidence scores, allowing systematic spurious correctness detection through confidence analysis.

3 Proposed Method

3.1 Overview

We propose a framework for fine-grained confidence estimation that enables LLMs to output confidence scores at the individual evidence triplet level. Given a question q, our framework produces (i) an answer a along with confidence score $c_a \in [0, 1]$, and (ii) a sequence of n evidence-confidence pairs $[(e_1, c_e^{(1)}), (e_2, c_e^{(2)}), ..., (e_n, c_e^{(n)})]$, where each e_i is a triplet composed of a subject, relation, and object (e.g., (*Tokyo Tower, height, 333m*)), and $c_e^{(i)} \in [0, 1]$.

To compute the confidence scores, we adopt two methods from Tian et al. (2023): (i) *model-based methods* (§3.2), which derive confidence from the model's intrinsic uncertainty during response generation, and (ii) *verbalized methods* (§3.3), which elicit self-reported confidence scores from the model via natural language prompts.

3.2 Model-based Methods

Given the question q, we estimate the conditional generation probabilities of the evidence triplets and the final answer, i.e., $p(e_1|q), p(e_2|q, e_1), ..., p(e_n|q, e_1, e_2, ..., e_{n-1})$ and $p(a|q, e_1, e_2, ..., e_n)$, in two ways and then use these probabilities as confidence scores.

First, *Token prob.* first prompts the model to generate the full reasoning sequence, including a sequence of evidence triplets and the final answer. For each component, we then extract the token-level probabilities associated with that component (e.g., $p(e_1^1|q), p(e_1^2|q, e_1^1)$, ... for the first evidence triplet), and compute the geometric mean of these token probabilities.

Second, *Label prob.* samples n reasoning sequences from the model. The final answers and sequences of evidence triplets are then separately grouped into clusters based on fuzzy matching,[2] and the most frequent cluster is selected as the final output. The confidence score for the final answer is the number of cluster elements divided by n. For evidence confidence, we select the evidence set E^* that appears most frequently among the n sampled trajectories, thereby preserving structural coherence. Each evidence triplet $e \in E^*$ is assigned a reliability score $p(e \mid q) = \frac{1}{n} \sum_{i=1}^{n} I[\, e \in E^{(i)}\,]$, which disentangles path-level coherence from the certainty of individual evidence pieces.

3.3 Verbalized Methods

Unlike model-based approaches, verbalized methods elicit confidence scores directly via prompting, using three variants.

First, *Verb. 1S* prompts the model to generate a sequence of evidence triplets and the final answer *along with* confidence scores in a single response. Second, *Verb. 1S CoT* first elicits CoT reasoning, then asks for confidence estimation. Third, *Ling. 1S* uses a similar prompt to Verb. 1S but replaces numerical scores with a 13-level linguistic scale (e.g., "almost certain," "likely") adapted from Fagen-Ulmschneider and translated into Japanese.

3.4 Prompt Design

To enable these confidence estimation methods, we design prompts that require models to simultaneously generate: (1) evidence as structured triplets in (Subject, Relation, Object) format, (2) confidence scores for each triplet, and (3) the final answer with its confidence score—all in a single forward pass to maintain contextual coherence. The evidence-first ordering and explicit confidence requirements for each component enable fine-grained uncertainty quantification. We include few-shot examples to ensure correct formatting and independent confidence evaluation. Full prompt templates are provided in Appendix Table 4[3].

4 Experimental Settings

This section describes our experimental setup, including the dataset, evaluation models, automated evaluation procedures, and metrics used to assess fine-grained confidence estimation performance.

4.1 Dataset

We conduct our experiments on JEMHopQA (Ishii et al., 2024b), a Japanese multi-hop QA benchmark whose training split contains 1,059 questions. We reserve 1,000 questions as our evaluation set and select three questions from the remaining 59 as

reasoning.

[2] We first normalize numerals and symbols, then merge lexically differing but semantically identical strings via fuzzy string matching using RapidFuzz (`https://github.com/maxbachmann/RapidFuzz`) with a fixed similarity threshold.

[3] Since our evaluation uses the Japanese JEMHopQA dataset (Ishii et al., 2024b), all prompts were originally designed in Japanese and translated to English for presentation.

few-shot exemplars for in-context prompting. Each question requires two to three reasoning hops, and the gold annotations provide, on average, 2.2 subject–relation–object triples as supporting evidence. Because these triple-level evidence annotations let us verify the correctness of every individual reasoning component, JEMHopQA is well suited for evaluating the validity of our proposed fine-grained confidence scores and for analysing spuriously correct answers whose evidence is partially erroneous.

4.2 Evaluation Models

We evaluate three models representing different training paradigms:

- **GPT-4.1-mini** (OpenAI, 2025) (ver. 2025-04-14, dense model likely incorporating human preference optimization; parameter count not publicly disclosed)
- **Llama4Maverick17B128E-InstructFP8** (Meta AI, 2025) (SFT+Instruct Mixture-of-Experts with 128 experts)[4]
- **Phi-4** (Abdin et al., 2024) (14B-parameter SFT-trained dense model)[5]

This diversity in architectures and training approaches demonstrates the generalizability of our method across different model types. We set the decoding temperature to 0.0 for all methods except *Label prob.*, which requires temperature 0.7 and top-p 0.95 for sampling-based confidence estimation (see §3.2). All experiments were conducted using the official APIs via Azure AIFoundry[6].

4.3 Evaluation Metrics

We evaluate our method along two dimensions: calibration and discrimination. For *calibration* metrics, following Tian et al. (2023), we report both raw and temperature-scaled scores.

For calibration, we use Expected Calibration Error (**ECE**; Guo et al., 2017), which is the average absolute difference between predicted confidence and actual accuracy across bins, and Brier Score (**BS**; BRIER, 1950), which is the mean squared difference between predicted probabilities and outcomes. Lower values indicate better calibration.

For discrimination, our metrics are:

AUC: Area under the selective accuracy-coverage curve (Geifman and El-Yaniv, 2017), measuring the ability to distinguish correct/incorrect predictions (higher is better).

ROC-AUC: Area under the Receiver Operating Characteristic curve (Fawcett, 2006) for spurious correctness detection (higher is better).

PR-AUC: Area under the Precision-Recall curve (Davis and Goadrich, 2006), particularly suitable for imbalanced spurious correctness detection (higher is better).

We also apply temperature scaling to calibrate confidence scores as $p' = \sigma(z/T)$ where $z = \log(p/(1-p))$, with the optimal temperature T found by 5-fold cross-validation minimizing ECE. Temperature-scaled metrics are denoted by "–t" (e.g., ECE-t, BS-t).

4.4 Automated Evaluation

All evaluation metrics require binary correctness labels for each answer and evidence triplet. We obtain these labels automatically with GPT-4.1 and then validate their reliability.

Answer evaluation: We use exact match for YES/NO questions (33% of the dataset). For entity-based questions (67%), including named entities, dates and numerical values, we employ GPT-4.1 to judge semantic equivalence when exact match fails.

Evidence evaluation: We instruct the model to perform one-to-one matching between predicted and gold-standard triplets, checking a set of matching conditions that tolerate surface-form variation (e.g. lexical paraphrase, subject–object swaps), assigning binary scores (1.0 or 0.0) to each pair.

Reliability assessment: To assess the reliability of automated evaluation, one of the authors manually labeled 100 randomly sampled instances per model (300 total). We then computed the agreement rates between these manual labels and the automatically assigned label. For answer correctness, the agreement rates were 98% (GPT-4.1-mini), 100% (Llama-4-Maverick), and 98% (Phi-4); for evidence correctness, they were 93%, 94%, and 95% respectively. While these agreement rates indicate that automated evaluation introduces small amount of noise into our measurements, it affects all compared methods equally, preserving the validity of relative performance comparisons.

[4] Azure internal model version 1; created Oct 1 2024, updated May 7 2025.

[5] Azure internal model version 7; created Oct 1 2024, updated Apr 16 2025.

[6] https://learn.microsoft.com/ja-jp/azure/ai-foundry/

Method	GPT-4.1-mini				Llama-4-Maverick				Phi-4			
	ECE↓	ECE-t↓	BS-t↓	AUC↑	ECE↓	ECE-t↓	BS-t↓	AUC↑	ECE↓	ECE-t↓	BS-t↓	AUC↑
Label prob.	**0.172**	0.139	**0.197**	**0.786**	**0.190**	0.145	**0.218**	**0.735**	**0.107**	**0.188**	**0.204**	**0.691**
Token prob.	0.264	0.135	0.245	0.650	–	–	–	–	–	–	–	–
Verb. 1S	0.297	0.125	0.210	0.791	0.316	0.137	0.243	0.702	0.508	0.326	0.329	0.574
Verb. 1S CoT	0.305	0.140	0.223	0.757	0.295	**0.086**	0.237	0.724	0.500	0.297	0.327	0.540
Ling. 1S	0.288	**0.054**	0.227	0.658	0.297	0.079	0.230	0.670	0.491	0.124	0.261	0.459

Table 1: Evidence confidence extraction performance at the triplet level. Bold indicates best performance, underline indicates worst performance. Label prob. consistently outperforms other methods across models.

5 Results

This section reports quantitative results based on the settings in Section 4, covering evidence confidence extraction methods (§5.1) and spurious correctness detection performance (§5.2). Comprehensive results for all confidence extraction methods across the three evaluated models are provided in Appendix Table 6.

5.1 Evidence Confidence Estimation

Table 1 presents the calibration and discrimination performance of different confidence extraction methods for evidence at the triplet level. Label prob. (frequency-based method with N=10 samplings, temperature 0.7, top-p 0.95) consistently achieves the best results across models, with ECE values ranging from 0.107 to 0.190.

Several key patterns emerge from these results. First, GPT-4.1-mini and MoE architectures (Llama-4-Maverick) show relatively good performance with verbalized methods, with temperature scaling proving particularly effective for reducing ECE (e.g., Verb. 1S CoT achieving ECE-t of 0.086 for Llama-4). In contrast, the smaller SFT model (Phi-4) shows poor performance with all verbalized methods (ECE > 0.5), suggesting that verbalized confidence expression requires sufficient model capacity. Despite this limitation, Phi-4's Label prob. performance remains competitive (ECE = 0.107), demonstrating the robustness of frequency-based approaches across model scales.

Fig. 2 visualizes these calibration results through reliability diagrams. The diagonal line represents perfect calibration where predicted confidence matches actual accuracy. Label prob. (left column) shows consistent near-diagonal performance across all models, confirming its superiority. While verbalized methods initially show poor calibration, temperature scaling dramatically improves their performance, as demonstrated by Llama-4-Maverick's Verb. 1S CoT achieving competitive calibration after scaling (bottom right).

Model	ROC-AUC / PR-AUC	
	Ans Conf.	Ev Conf.
GPT-4.1-mini	0.59 / 0.46	0.74 / 0.56
Llama-4-Maverick	0.53 / 0.37	0.69 / 0.55
Phi-4	0.65 / 0.63	0.84 / 0.82

Table 2: Spurious correctness detection performance using Label prob. Evidence confidence consistently outperforms answer confidence across all models.

5.2 Spurious Correctness Detection

Building on the evidence confidence results, we evaluate how effectively these confidence scores can detect spurious correctness—cases where answers are correct but reasoning is flawed.

For detection, we aggregate triplet-level confidence scores by taking the minimum value across all evidence triplets, reflecting that reasoning validity requires all evidence to be correct.[7]

Table 2 summarizes detection performance across models using Label prob. method, which demonstrated the best calibration in §5.1.

Evidence confidence consistently provides superior discrimination compared to answer confidence, with Phi-4 achieving the highest ROC-AUC of 0.84 despite being the smallest model. This exceptional performance motivates a closer examination of how confidence scores distribute for different correctness patterns.

Fig. 3 visualizes the relationship between answer and evidence confidence for Phi-4's Label prob. method, revealing how spurious correctness cases can be identified through confidence patterns.

The scatterplot reveals distinct patterns: spurious correctness cases (blue) concentrate in the upper-left region where evidence confidence is low ($c_e < 0.3$) but answer confidence remains high ($c_a > 0.8$). This separation enables effective detection using evidence confidence as a discriminator.

[7] We also evaluated mean aggregation, which showed comparable but slightly inferior performance, particularly for PR-AUC.

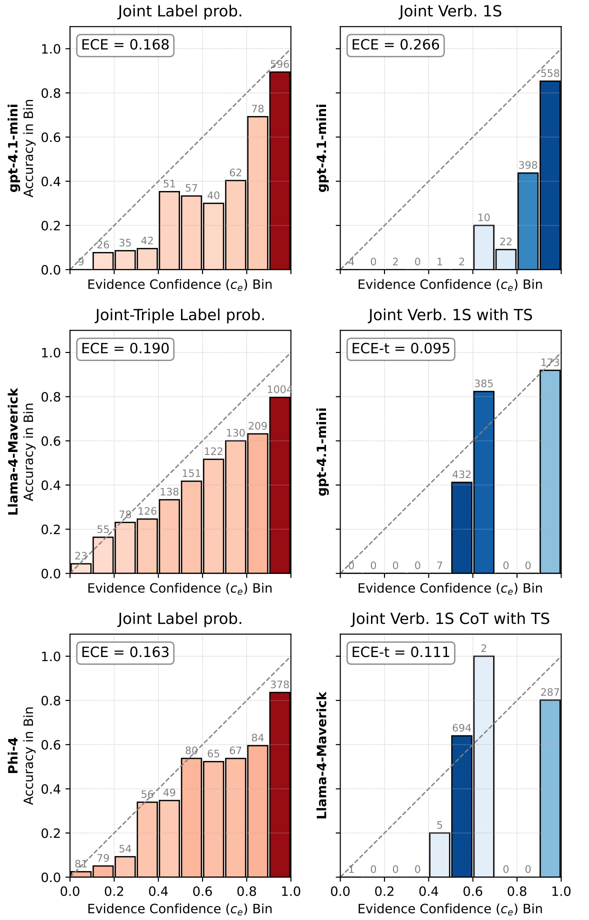

Figure 2: Reliability diagrams for evidence confidence calibration. Label prob. (left) shows consistent calibration across all models. Temperature scaling dramatically improves verbalized methods (right), with Llama-4-Maverick's Verb. 1S CoT achieving the best calibration after scaling (ECE-t=0.086).

The quantitative effectiveness of this approach is further demonstrated through ROC and PR curves in Appendix Fig. 5.

6 Analysis

This section analyzes the improvement in answer confidence calibration through joint generation (§6.1) and patterns in evidence confidence errors (§6.2).

6.1 Answer Confidence Calibration Improvement

A natural hypothesis emerges from our approach: by explicitly requiring models to assess evidence confidence, we might encourage more careful reasoning, potentially leading to better-calibrated answer confidence as well. In other words, does the very act of evaluating evidence confidence indeed improve the model's ability to assess its own answer confidence?

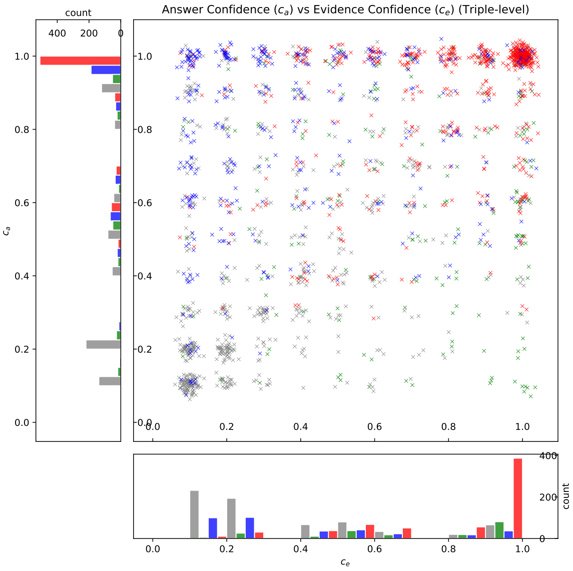

Figure 3: Answer confidence vs evidence confidence scatter plot (Label prob.). **Red**: Both answer and evidence correct (true correct), **Blue**: Answer correct but evidence wrong (spurious correctness), **Green**: Evidence correct but answer wrong, **Gray**: Both answer and evidence wrong. The histograms show marginal distributions, revealing that spurious correctness cases (blue) cluster at low evidence confidence.

Model	Method	Only-Answer ECE/AUC	Joint-answer ECE/AUC	Improv. Rate ECE/AUC
GPT-4.1-mini	Label prob.	0.23/0.73	0.17/0.84	26%/16%
Llama-4-Maverick	Verb. 1S	0.42/0.55	0.24/0.77	43%/40%
Phi-4	Label prob.	0.14/0.68	0.16/0.75	-16%/11%

Table 3: Answer confidence performance: ECE and AUC values for answer-only vs. joint generation approaches. Lower ECE indicates better calibration; higher AUC indicates better discrimination. Improvement rates show the relative change from answer-only to joint generation.

Our results confirm this hypothesis. Table 3[8] shows that joint generation of answer and evidence confidence substantially improves answer confidence calibration in most cases, with ECE reductions of 26% and 43% for GPT-4.1-mini and Llama-4-Maverick models respectively. Moreover, AUC

[8] Only-answer and joint generation prompts are provided in Appendix Tables 5 and 4.

improvements range from 11% to 40% across three models, demonstrating enhanced discrimination capability.

The improvement is particularly pronounced for GPT-4.1-mini (26% ECE reduction, 16% AUC improvement) and Llama-4-Maverick (43% ECE reduction, 40% AUC improvement). GPT-4.1-mini, likely optimized for human preferences, tend toward overconfidence in answer-only settings. Joint generation appears to mitigate this by forcing explicit reasoning about evidence uncertainty. The improvement is the largest for Llama-4-Maverick, possibly because different experts can specialize in answer versus evidence generation, leading to more nuanced confidence expressions. Phi-4's ECE worsened (-16%), which may reflect its already-low baseline ECE (0.14) leaving less room for improvement. However, the 11% AUC improvement shows that joint generation still enhances error detection capability.

The proposed method of jointly estimating answer and evidence confidence improved not only ECE (better calibration between predicted confidence and actual accuracy) but also AUC (better discrimination between correct and incorrect predictions) in almost all settings (see Fig. 4 for visual comparison). The improvement is particularly notable because it demonstrates that generating evidence alongside answers helps the model better calibrate its answer confidence—even though we might expect the additional complexity to potentially harm calibration.

The consistent improvements across models suggest that requiring explicit evidence assessment fundamentally changes how models evaluate their own certainty. By forcing models to decompose reasoning into verifiable components and assign confidence to each, we create a more structured uncertainty quantification process. Our ablation study (Appendix D) confirms that both evidence generation and explicit confidence scoring contribute to this improvement, with evidence generation alone improving answer accuracy by 6.8-13.8% and additional confidence requirements further enhancing calibration. This joint generation maintains full context while preventing the overconfidence often observed in answer-only generation, where models lack explicit mechanisms to surface intermediate uncertainties. The importance of maintaining unified context is further confirmed by our preliminary experiments (Appendix E), where separating generation steps degraded performance significantly (e.g., answer confidence AUC dropping from 0.848 to 0.722).

6.2 Evidence Confidence Error Analysis

We analyzed error patterns in Label prob. results across three models, examining cases where confidence scores misalign with correctness. We extracted 30 samples per model (90 samples in total) for two critical patterns: high confidence despite incorrect evidence and low confidence despite correct evidence.

6.2.1 High Confidence for Incorrect Evidence

We examined 90 cases where models assigned maximum confidence ($c_e = 1.0$) to incorrect evidence triplets, revealing four primary error patterns (see Appendix Table 9 for detailed distribution):

Numerical/Temporal Drift (49%): Nearly half of high-confidence errors involve values numerically close to correct answers. The model assigns full confidence to values it considers numerically "close enough", such as neighbouring years (1873 vs. 1871) or small miscounts (12 cities vs. 14 cities). Such drift occurs mainly for ages, counts, and areas, whereas high-precision temporal facts that require an exact calendar date (e.g. 17 May 1964) usually receive lower confidence.

Entity Conflation (38%): Models confidently substitute entities with similar names or shared categories. This systematic confusion in entity disambiguation allows surface-level similarities to override factual distinctions, particularly affecting person names, company names, and locations.

Question-Answer Contamination (10%): Models exhibit a copy-paste bias, directly transferring values from questions into evidence triplets. For example, given "Which of City A or City B has azalea as its city flower?", models generate high-confidence triplets like (City A, city flower, azalea) regardless of factual accuracy.

Default Value Bias (2%): Though less frequent, models occasionally apply statistical priors with high confidence, such as assuming March 31st as the end of a fiscal year—a default particularly common in our dataset, reflecting training data patterns specific to Japanese business context.

6.2.2 Low Confidence for Correct Evidence

Analysis of 90 correct triplets with low confidence ($0.1 \leq c_e \leq 0.4$ for GPT-4.1-mini/Llama-4; $0.1 \leq c_e \leq 0.3$ for Phi-4) reveals that conservative confidence often reflects legitimate uncertainty

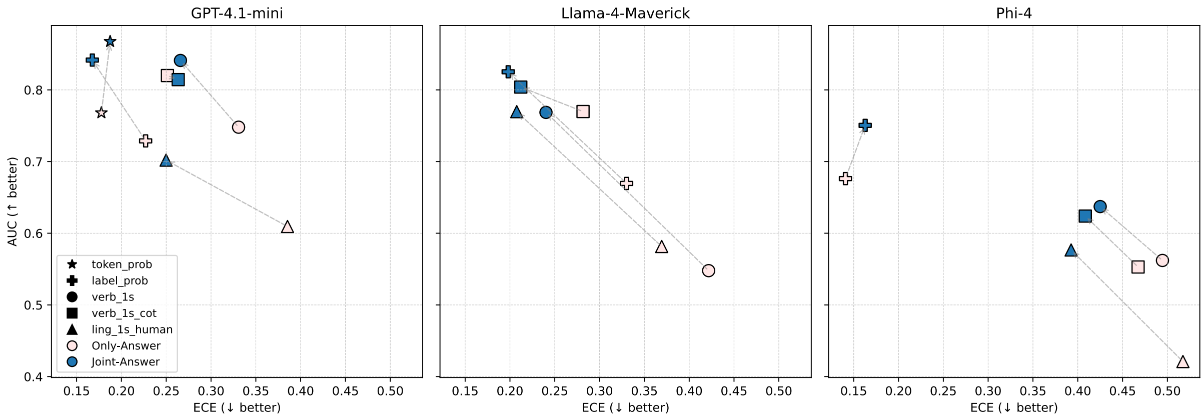

Figure 4: Plot of answer confidence for the baseline Answer-only method versus the Joint-Answer method (simultaneous evidence generation) across all models.

(detailed breakdown in Appendix Table 10):

Competing Plausible Alternatives (27%): Models reduce confidence when multiple valid candidates exist. For instance, when generating Don Shirley's birthplace, near-equal sampling of "United States" (correct), "Berlin", and "New York City" results in low confidence due to competing claims in the training data.

Complex Relation Mapping (22%): Confidence decreases when relations embody multi-hop compressions (e.g., "singer of a theme song (of something)") or ambiguous question-to-triplet mapping (e.g., "Did both A and B complete graduate school?" leading to different educational status representations).

Date/Numerical Values (21%): Specific dates and large numbers receive low confidence even when correct, demonstrating appropriate epistemic humility about precise numerical facts.

Surface Form Variations (11%): Equivalent expressions (e.g., "18+" vs. "CERO D" for age ratings) reduce confidence due to our automated evaluation's exact match limitations rather than genuine model uncertainty.

Rare/Long-tail Entities (10%): Information about local mascots or other infrequent facts receives conservative confidence scores.

Multi-valued Relations (9%): Relations with multiple valid values (e.g., "neighboring cities") trigger lower confidence as probability mass distributes across alternatives.

These patterns reveal the tendency that high-confidence errors arise when the model assigns a high probability to the incorrect answers that are semantically close to the correct ones (e.g., adjacent years, near-duplicate entity names), presumably because those expressions occupy neighboring regions in the model's internal representation, while low-confidence errors reflect the situations in which multiple answers are equally plausible or genuinely unknown, so the model spreads its probability mass across them and gives any single candidate a low score. Given that LLMs represent knowledge in a continuous space and fundamentally operate on probabilistic principles, such phenomena may be inevitable. Nevertheless, our results suggest that a key challenge lies in finely discriminating between subtly different facts within this latent space, while preserving the robustness of knowledge processing to reduce overconfidence.

7 Conclusion

This paper introduced a fine-grained confidence estimation framework that extends LLM uncertainty quantification from answer-level to individual evidence components. By decomposing reasoning into triplets and assigning confidence scores to each component, we enabled precise error detection within reasoning chains, a capability absent from existing coarse-grained approaches.

Future work should explore alternative evidence decomposition strategies beyond triplet format, investigate the relationship between granularity and confidence quality, and extend evaluation to other languages and reasoning tasks. As LLMs increasingly support high-stakes decisions, fine-grained confidence estimation will be essential for trustworthy deployment.

Limitations

While our results demonstrate the effectiveness of fine-grained confidence estimation, several limitations warrant discussion:

Automated evaluation reliability: While our automated evaluation achieved high agreement with human judgments (93-100% across different models and metrics), this approach has inherent limitations. The reliability may vary with different model families or task complexities not tested in our validation. Furthermore, our validation sample of 100 instances per model may not capture all edge cases. Future work should explore more robust evaluation methods, potentially combining multiple evaluators or using specialized evaluation models.

Dataset and language specificity: Our evaluation focused on Japanese multihop QA. While the underlying principles should transfer to other languages and tasks, empirical verification is needed.

Evidence format constraints: We used triple-format evidence (Subject, Relation, Object), which works well for factual QA but may not suit all reasoning types. Future work should explore other evidence representations.

Computational tradeoffs: While our method is more efficient than extensive resampling approaches, it still requires generating additional tokens for evidence and confidence. Future work could explore more efficient confidence estimation methods.

Calibration versus discrimination tradeoff: While we generally see improvements in both metrics, some configurations show tension between calibration and discrimination performance. Understanding and optimizing this tradeoff remains an open challenge.

Acknowledgments

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

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

GLENN W. BRIER. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Wade Fagen-Ulmschneider. Perception of Probability Words. https://waf.cs.illinois.edu/visualizations/Perception-of-Probability-Words/.

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

Yonatan Geifman and Ran El-Yaniv. 2017. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Ai Ishii, Naoya Inoue, Hisami Suzuki, and Satoshi Sekine. 2024a. Analysis of LLM's "spurious" correct answers using evidence information of multi-hop QA datasets. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 24–34, Bangkok, Thailand. Association for Computational Linguistics.

Ai Ishii, Naoya Inoue, Hisami Suzuki, and Satoshi Sekine. 2024b. JEMHopQA: Dataset for Japanese explainable multi-hop question answering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9515–9525, Torino, Italia. ELRA and ICCL.

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.

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. *Preprint*, arXiv:2503.15850.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfcheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Meta AI. 2025. Llama 4 maverick 17b-128e instruct. Model release date: April 5, 2025. Llama 4 Maverick is a 17B parameter, 128-expert, natively multimodal large language model released under the Llama 4 Community License. Knowledge cutoff: August 2024.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

OpenAI. 2025. Introducing gpt-4.1 in the api. Announces the release of GPT-4.1, GPT-4.1 mini, and GPT-4.1 nano, with major improvements in coding, instruction following, and long context handling.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in llms. *Preprint*, arXiv:2502.06233.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

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.

A Prompt Templates

The following tables present the prompt templates used in our experiments. Table 4 shows the prompts for our main joint generation approach, while Table 5 contains the prompts for the answer-only baseline used in the ablation study (§6.1) to demonstrate the improvement from joint evidence-confidence generation. As our evaluation was conducted on the Japanese JEMHopQA dataset, all prompts were originally written in Japanese and have been translated to English for this presentation. The actual experiments used the Japanese versions of these prompts.

B Detailed Experimental Results

Table 6 presents the complete experimental results for all confidence extraction methods across the three evaluated models. For each model and method combination, we report both answer and evidence confidence performance metrics. The table shows that Label prob. consistently achieves the best overall performance across models, particularly for evidence confidence calibration.

C Additional Experimental Analysis

C.1 Spurious Correctness Detection Performance

Fig. 5 provides a detailed visualization of spurious correctness detection performance, showing both ROC and PR curves for the best-performing configuration (Phi-4 with Label prob.). The substantial gap between evidence confidence (orange) and answer confidence (blue) demonstrates that fine-grained confidence at the evidence level provides significantly better discrimination for identifying cases where correct answers are supported by incorrect reasoning.

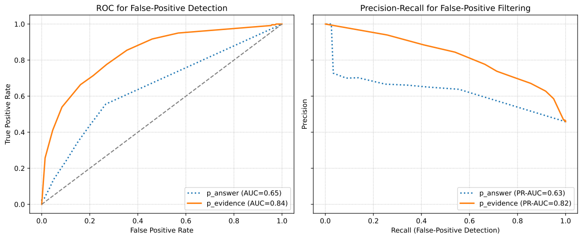

Figure 5: ROC and PR curves for spurious correctness detection using Phi-4/Label prob. . Evidence confidence (orange) achieves ROC-AUC 0.84 and PR-AUC 0.82, significantly outperforming answer confidence (blue) with ROC-AUC 0.65 and PR-AUC 0.63.

D Ablation Study on Evidence and Confidence Generation

To investigate the mechanism behind the calibration improvements observed in our main experiments, we conducted an ablation study with four configurations:

<table>
<tr><th>Method</th><th>Template (Joint)</th></tr>
<tr><td>Label prob. /
Token prob.</td><td>Provide an answer to the question and the supporting evidence as triples. Triples should be in the format (Subject, Relation, Object). Subject is an entity, Object is an entity or concrete value (date, number, etc.), both as short single phrases.
Output in the following format:
Triple1: (Subject, Relation, Object)
Triple2: (Subject, Relation, Object)
...
Answer: YES|NO|<short single phrase>
Example:
Q: Which console had more games launch simultaneously with its hardware in Japan, the Wii U or GameCube?
Triple1: (Wii U, simultaneous Japan launch titles, 11)
Triple2: (GameCube, simultaneous Japan launch titles, 2)
Answer: Wii U
[2 more examples omitted]
Question: {THE_QUESTION}</td></tr>
<tr><td>Verb. 1S</td><td>Provide an answer to the question and the supporting evidence as triples.
Triples should be in the format (Subject, Relation, Object). Subject is an entity, Object is an entity or concrete value (date, number, etc.), both as short single phrases.
Include a confidence score (0.00-1.00, two decimals) for each triple and for the final answer.
Output in the following format:
Triple1: (Subject, Relation, Object) 0.00-1.00
Triple2: (Subject, Relation, Object) 0.00-1.00
...
Answer: YES|NO|<short single phrase> 0.00-1.00
[3 more examples omitted]
Question: {THE_QUESTION}</td></tr>
<tr><td>Verb. 1S CoT</td><td>Provide an answer to the question and the supporting evidence as triples.
First show your thinking process concisely, then output the evidence triples and answer.
Triples should be in the format (Subject, Relation, Object). Subject is an entity, Object is an entity or concrete value (date, number, etc.), both as short single phrases.
Include a confidence score (0.00-1.00, two decimals) for each triple and for the final answer.
Output in the following format:
Thought: [reasoning process]
Triple1: (Subject, Relation, Object) 0.00-1.00
Triple2: (Subject, Relation, Object) 0.00-1.00
...
Answer: YES|NO|<short single phrase> 0.00-1.00
Example:
Q: Which console had more games launch simultaneously with its hardware in Japan, the Wii U or GameCube?
Thought: Need to compare the number of launch titles for both consoles in Japan.
Triple1: (Wii U, simultaneous Japan launch titles, 11) [confidence]
Triple2: (GameCube, simultaneous Japan launch titles, 2) [confidence]
Answer: Wii U [confidence]
[2 more examples omitted]
Question: {THE_QUESTION}</td></tr>
<tr><td>Ling. 1S</td><td>Provide an answer to the question and the supporting evidence as triples.
First show your thinking process, then output the evidence triples and answer.
Triples should be in the format (Subject, Relation, Object).
Subject is an entity, Object is an entity or concrete value (date, number, etc.), both as short single phrases.
Express confidence for each triple and the answer using one of these expressions:
{EXPRESSION_LIST}
Output in the following format:
Triple1: (Subject, Relation, Object) Almost certain|Very likely|...|Almost impossible
Triple2: (Subject, Relation, Object) Almost certain|Very likely|...|Almost impossible
...
Answer: YES|NO|<short single phrase> Almost certain|Very likely|...|Almost impossible
Question: {THE_QUESTION}</td></tr>
</table>

Table 4: Prompt templates for Joint configuration. All methods require evidence generation before the final answer. {THE_QUESTION} is replaced with the actual question, and {EXPRESSION_LIST} contains the 13-level linguistic expressions.

Method	Template (Only-Answer)
Label prob. / Token prob.	Provide the best answer to the following question. Output ONLY the answer, without any other words or explanations. Question: {THE_QUESTION} Answer in the following format: Answer: <most likely answer, as short as possible; not a complete sentence, just the answer!>
Verb. 1S	Provide the best answer to the following question and the probability that it is correct (0.0 to 1.0). Output ONLY the answer and probability, without any other words or explanations. Question: {THE_QUESTION} Answer in the following format: Answer: <most likely answer, as short as possible; not a complete sentence, just the answer!> Probability: <probability your answer is correct (between 0.0 and 1.0), no additional comments; just the probability!>
Verb. 1S CoT	Show your step-by-step thinking process for the following question. Then provide the answer and the probability that it is correct (0.0 to 1.0). Question: {THE_QUESTION} Answer in the following format: Thought: <explain your thinking process in one concise sentence> Answer: <most likely answer, as short as possible; not a complete sentence, just the answer!> Probability: <probability your answer is correct (between 0.0 and 1.0), no additional comments; just the probability!>
Ling. 1S	Provide the best answer to the following question and express your confidence using one of these expressions: {EXPRESSION_LIST} Question: {THE_QUESTION} Answer in the following format: Answer: <most likely answer, as short as possible; not a complete sentence, just the answer!> Answer Confidence: <confidence expression, no additional comments; just the short phrase!>

Table 5: Prompt templates for Only-Answer configuration. {THE_QUESTION} is replaced with the actual question, and {EXPRESSION_LIST} contains the 13-level linguistic expressions adapted from Fagen-Ulmschneider.

Model	**Method**	**Joint Answer**						**Joint Evidence**					
		Accuracy	**ECE↓**	**ECE-t↓**	**BS↓**	**BS-t↓**	**AUC↑**	**Accuracy**	**ECE↓**	**ECE-t↓**	**BS↓**	**BS-t↓**	**AUC↑**
GPT-4.1-mini	Label prob.	0.670	0.168	0.132	0.167	0.160	0.842	0.621	0.172	0.139	0.221	0.197	0.786
	Verv. 1S	0.654	0.266	0.095	0.272	0.202	0.841	0.629	0.297	0.125	0.300	0.210	0.791
	Verv. 1S CoT	0.667	0.263	0.086	0.274	0.207	0.814	0.627	0.305	0.140	0.312	0.223	0.757
	Ling. 1S	0.670	0.250	0.065	0.270	0.212	0.702	0.614	0.288	0.054	0.307	0.227	0.658
	Token prob.	0.676	0.187	0.108	0.188	0.159	0.867	0.645	0.264	0.135	0.313	0.245	0.650
Llama-4-Maverick	Label prob.	0.656	0.198	0.152	0.193	0.180	0.825	0.611	0.190	0.145	0.245	0.218	0.735
	Verv. 1S	0.660	0.240	0.115	0.252	0.222	0.769	0.601	0.316	0.137	0.317	0.243	0.702
	Verv. 1S CoT	0.679	0.212	0.111	0.232	0.217	0.804	0.614	0.295	0.086	0.302	0.237	0.724
	Ling. 1S	0.685	0.208	0.075	0.236	0.198	0.770	0.586	0.297	0.079	0.309	0.230	0.670
Phi-4	Label prob.	0.526	0.163	0.175	0.191	0.205	0.750	0.449	0.107	0.188	0.178	0.204	0.691
	Verv. 1S	0.543	0.425	0.244	0.418	0.289	0.637	0.459	0.508	0.326	0.495	0.329	0.574
	Verv. 1S CoT	0.554	0.408	0.204	0.407	0.294	0.624	0.463	0.500	0.297	0.491	0.327	0.540
	Ling. 1S	0.541	0.393	0.027	0.396	0.247	0.577	0.446	0.491	0.124	0.486	0.261	0.459

Table 6: Comprehensive results for all confidence extraction methods. Bold values indicate best performance for each metric within each model group.

Configuration	GPT-4.1-mini		Llama-4-Maverick		Phi-4	
	Accuracy	ECE ↓	Accuracy	ECE ↓	Accuracy	ECE ↓
C1: Answer only + conf.	0.528	0.363	0.544	0.422	0.473	0.495
C2: Answer + Evidence, no conf.	0.666	—	0.650	—	0.541	—
C3: Answer + Evidence, answer conf. only	0.650	0.280	0.659	0.326	0.526	0.440
C4: Answer + Evidence + both conf.	0.654	0.266	0.660	0.240	0.542	0.426

Table 7: Ablation study on incremental effects of evidence and confidence generation using the Verb. 1S method. ECE values are not applicable for C2 as no confidence scores are generated.

- **C1:** Answer only with confidence (baseline)
- **C2:** Answer + Evidence, no confidence scores
- **C3:** Answer + Evidence, answer confidence only
- **C4:** Answer + Evidence, both answer and evidence confidence (our full method)

Table 7 reveals two key findings:

Evidence generation improves accuracy: Comparing C1 to C2, we observe substantial accuracy improvements across all models (GPT-4.1-mini: +13.8%, Llama-4-Maverick: +10.6%, Phi-4: +6.8%), confirming that explicit evidence generation enhances reasoning.

Evidence confidence scoring improves answer calibration: Comparing C3 to C4, adding evidence confidence requirements consistently improves answer confidence calibration (ECE reduction: GPT-4.1-mini: 0.280→0.266, Llama-4-Maverick: 0.326→0.240, Phi-4: 0.440→0.426).

The minor variations in accuracy between C2, C3, and C4 suggest that confidence scoring itself does not significantly impact answer correctness, but rather improves calibration through more realistic uncertainty expressions.

E Preliminary Experiments on Generation Strategies

To validate our joint generation approach, we conducted preliminary experiments comparing three generation strategies on 120 samples from the JEMHopQA development set:

- **Joint generation (verb_1s)**: Generate answer, evidence, and confidence scores in a single response
- **Sequential dialogue (verb_2s)**: Generate answer and evidence first, then request confidence scores in the same message
- **Independent steps**: Generate confidence scores in a separate message

Table 8 shows that maintaining unified context throughout the generation process is crucial for accurate confidence estimation. Even the sequential approach within the same message shows performance degradation compared to joint generation, suggesting that the model benefits from considering confidence while generating the content itself.

	Answer Confidence			Evidence Confidence		
Method	ECE-t↓	BS-t↓	AUC↑	ECE-t↓	BS-t↓	AUC↑
Joint (Verb. 1S	0.113	0.180	0.848	0.101	0.199	0.731
Sequential (Verb. 2S	0.119	0.184	0.766	0.130	0.204	0.692
Independent	0.263	0.230	0.722	0.246	0.266	0.672

Table 8: Performance comparison of generation strategies. Joint generation consistently outperforms separated approaches, with the degradation being most severe when confidence is generated in an independent message.

Note: These preliminary experiments used a smaller dataset and slightly different evaluation criteria than the main experiments, hence the absolute numbers differ from those reported in the main text.

F Detailed Error Analysis Tables

The following tables provide detailed breakdowns of the error patterns observed in our analysis of confidence misalignment cases.

Error Type	GPT-4.1-mini	Llama-4	Phi-4	Total (%)
Numerical/ Temporal Drift	16	14	14	44 (49%)
Entity Conflation	8	14	11	34 (38%)
Question-Answer Contamination	4	2	4	9 (10%)
Default Value Bias	1	0	1	2 (2%)
Insufficient Granularity	1	0	0	1 (1%)

Table 9: Distribution of error types in high-confidence incorrect evidence (n=90, 30 samples per model). All cases exhibited maximum confidence ($c_e = 1.0$).

Pattern	GPT-4.1-mini	Llama-4	Phi-4	Total (%)
Competing Plausible Alternatives	9	6	9	24 (27%)
Complex Relation Mapping	5	9	6	20 (22%)
Numerical Values	11	4	4	19 (21%)
Surface Form Variations	2	4	4	10 (11%)
Rare/Long-tail Entities	0	2	7	9 (10%)
Multi-valued Relations	3	5	0	8 (9%)

Table 10: Distribution of patterns in low-confidence correct evidence (n=90, 30 samples per model).