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

Ai Ishii[1,2], Naoya Inoue[2], Hisami Suzuki[3], Satoshi Sekine[3]
[1]BIPROGY Inc., Tokyo, Japan,
[2]Japan Advanced Institute of Science and Technology, Ishikawa, Japan,
[3]National Institute of Informatics, Tokyo, Japan
ai.ishii@jaist.ac.jp, naoya-i@jaist.ac.jp, hisamis@nii.ac.jp, sekine@nii.ac.jp

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. Using carefully designed prompts, we generate 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 both Japanese and English multi-hop QA benchmarks across multiple models from three model families representing different architectures and training approaches, we show that our approach exhibits superior calibration performance for evidence confidence and demonstrates effective ability to detect spurious correct answers (up to 0.84 on our primary discrimination metric). The consistent improvements across languages demonstrate the generalizability of our method. 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

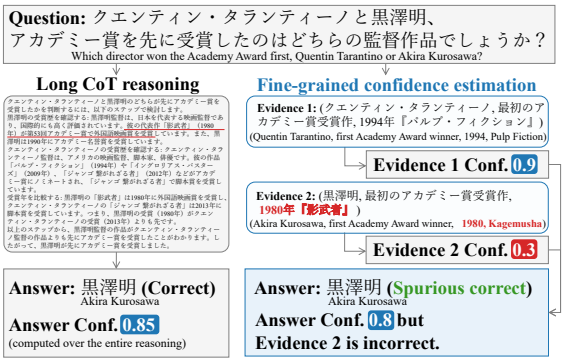

Figure 1: Overview: Fine-grained triplet-based confidence scores enable precise error localization (e.g., incorrect year with confidence 0.3), unlike coarse-grained confidence that masks specific mistakes within reasoning chains.

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 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 ques-

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1238–1257

tion (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. Unlike prior confidence calibration over whole chains or free-text claims, we operate at the triplet level: (Subject, Relation, Object) evidence units jointly generated with the answer. For confidence extraction, we adapt 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 multi-hop 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 at the evidence triplet level for multi-hop reasoning. 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 up to 0.84.

We release code and JP/EN prompts to facilitate reproduction.[1]

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. Tian et al. (2023) observed that probability distributions can be affected by human preference optimization (HPO) in certain model configurations.

Linguistic confidence expression: Tian et al. (2023) demonstrated that prompting models to self-report their confidence produces better-calibrated scores than relying on token probabilities alone, particularly in HPO-trained models. Confidence can be expressed either as explicit numerical probabilities or as qualitative phrases (e.g., "almost certain"or "likely").

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

The relative effectiveness of these approaches depends on model architecture and task characteristics. Importantly, none of these methods provide confidence scores at a granularity that identifies 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 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.

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.

Closely related, Zhang et al. (2024) introduce calibration over free-text "atomic" claims in long-form outputs. While their approach achieves fine-grained calibration, the free-text format can

[1] https://github.com/aiishii/finegrained_conf

make deterministic alignment to gold evidence and step-specific error localization non-trivial in many settings. In contrast, we elicit generation-time confidence on (Subject, Relation, Object) triplets jointly with the answer, enabling one-to-one alignment to gold evidence and real-time assessment. This structured unit preserves relation-aware hop dependencies and demonstrates progressive improvements with finer granularity in spurious-correctness detection—without retraining.

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.[2] In this work, we 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.

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

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 after strong normalization,[3] 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 $p(e \mid q) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[\, e \in E^{(i)} \,]$, 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]We first normalize numerals and symbols (e.g., full-/half-width unification), remove non-essential parenthetical segments and punctuation, collapse whitespace, lowercase, and convert Japanese numerals to Arabic. Semantically identical strings that become identical after this normalization are merged.

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 5.[4]

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. Additional implementation details, hyperparameter settings, and reproducibility information are provided in Appendix A.1.

4.1 Dataset

We conduct our main 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 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.

For example, given the question "クエンティン・タランティーノと黒澤明、アカデミー賞を先に受賞したのはどちらの監督作品でしょうか？" (Which director won the Academy Award first, Kurosawa or Tarantino?), the gold evidence includes triplets such as (黒澤明/Akira Kurosawa, 初アカデミー賞受賞年/year of first Academy Award, 1951) and (タランティーノ/Quentin Tarantino, 初アカデミー賞受賞年, 1994), leading to the answer "黒澤明/Akira Kurosawa". This structured format enables precise evaluation of each reasoning component.

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.

To confirm language generality, we further applied our pipeline to 2WikiMultiHopQA (Ho et al., 2020) (English, 300 examples). Both datasets provide gold evidence in triplet format, allowing consistent evaluation across languages.

4.2 Baseline Model

As a chain-level baseline model, we collapse all evidence triplets into a single reasoning unit and obtain a single confidence score verbalized by the model for the entire chain. We choose verbalized confidence (rather than sampling-based aggregation) because variable-length, free-form chains make clustering-based estimation impractical, in contrast to short, structured outputs (answers or triplets). This baseline model provides a granularity ladder—answer-only → chain-level → triplet-level—used in our comparisons. Prompt templates for this baseline model and other settings are listed in Appendix Table 5.

4.3 Models

We evaluate the following models spanning different scales, training paradigms, and architectures:[5]

JEMHopQA (Japanese):

- **GPT–4.1–mini** (OpenAI, 2025) (ver. 2025–04–14; dense model, likely HPO)
- **GPT–4.1–nano** (OpenAI, 2025) (ver. 2025–04–14)
- **Llama-4-Maverick-17B-128E-Instruct-FP8** (Meta AI, 2025) (SFT + instruction-tuned Mixture-of-Experts with 128 experts)[6]
- **Phi–4** (Abdin et al., 2024) (14B-parameter SFT-trained dense model)[7]

2WikiMultiHopQA (English):

[4]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.

[5]The number of model parameters for GPT-4.1 variants is not publicly disclosed.

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

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

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↑
Baseline (Chain)	0.295	0.064	0.225	0.728	0.285	0.093	0.232	0.712	0.516	0.337	0.348	0.545
Label prob.	0.172	0.139	0.197	0.781	**0.190**	0.145	**0.218**	**0.733**	**0.107**	**0.188**	**0.204**	**0.695**
Token prob.	**0.095**	0.072	**0.186**	**0.816**	–	–	–	–	–	–	–	–
Verb. 1S	0.297	0.125	0.210	0.791	0.316	0.137	0.243	0.710	0.508	0.326	0.329	0.574
Verb. 1S CoT	0.305	0.140	0.223	0.754	0.295	0.086	0.237	0.716	0.500	0.297	0.327	0.549
Ling. 1S	0.288	**0.054**	0.227	0.669	0.297	**0.079**	0.230	0.652	0.491	0.124	0.261	0.457

Table 1: Evidence confidence extraction performance at the triplet level. Baseline (Chain) represents collapsed chain-level confidence. Bold indicates best performance per model. Token prob. shows model-dependent calibration (best for GPT-4.1-mini, worse than Label prob. for GPT-4.1-nano shown in Table 8), while Label prob. achieves consistently robust performance across all models (ECE 0.107-0.190, AUC 0.695-0.781). (Token prob. not evaluated for Llama-4-Maverick/Phi-4 since Azure AI Foundry Serverless doesn't expose token-level probs.)

- **GPT–4.1** (OpenAI, 2025) (ver. 2025–04–14)
- **GPT–4.1–mini**
- **Llama-4-Maverick-17B-128E-Instruct-FP8**

This diversity in model scales, architectures, and training approaches demonstrates the generalizability of our method across model types. We set the decoding temperature to 0.0 for all methods except *Label prob.*, which uses temperature 0.7 and top–p 0.95 for sampling–based confidence estimation (see §3.2); all experiments were conducted via the official APIs on Azure AI Foundry[8]

4.4 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).

4.5 Automated Evaluation

We obtain binary correctness labels using GPT-4.1 via a constrained function-calling interface (temperature=0.0). For evidence–gold scoring, the model returns a JSON results array following our rubric; prompt templates are in Table 4 and further details in Appendix A.1. In our evaluation set, YES/NO questions account for 33% and entity questions for 67%; we assess answer correctness accordingly (exact match for YES/NO, semantic equivalence for entities). Manual validation on 300 samples showed 93–100% agreement with human judgments (Appendix A.2).

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). We also briefly summarize cross-lingual validation and model size effects, with detailed results provided in Appendix B.1.1 and B.2. Comprehensive results for all confidence extraction methods across all evaluated models are in Appendix Table 8.

5.1 Evidence Confidence Estimation

Table 1 presents the calibration and discrimination performance of different confidence extraction methods for evidence at the triplet level. Among model-based methods, both Label prob. and Token prob. achieve strong calibration, though with

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

model-dependent patterns: Token prob. excels for GPT-4.1-mini (ECE 0.095) but underperforms for GPT-4.1-nano shown in Table 8 (ECE 0.140 vs Label's 0.096) , indicating sensitivity to model characteristics. In contrast, Label prob. (frequency-based method with N=10 samplings, temperature 0.7, top-p 0.95) demonstrates consistent performance across all models (ECE 0.096-0.190), making it more reliable for deployment where model selection may vary.

Table 1 also includes a chain-level baseline, which assigns one confidence score per reasoning chain. Triplet-level estimation yields monotonically higher AUC and lower ECE, demonstrating the benefit of finer-grained uncertainty modeling.

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. 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 the comparison between Label prob. and Token prob. through reliability diagrams. The diagonal line represents perfect calibration where predicted confidence matches actual accuracy. The left column for all models, plus bottom right for Phi-4 show Label prob.'s consistent near-diagonal performance across all models (nano: 0.096, mini: 0.172, Llama-4: 0.190, Phi-4: 0.107). The top and middle right for GPT-4.1 variants show Token prob., revealing model-dependent behavior: while achieving lower ECE than Label prob. for mini (0.095 vs 0.172), it shows higher ECE for nano (0.140 vs 0.096).

We further confirmed that the same calibration trends hold for the English 2WikiMultiHopQA dataset, where Label prob. maintains the lowest ECE among all methods, demonstrating the cross-lingual robustness of our confidence estimation framework (Appendix B.1.1).

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.

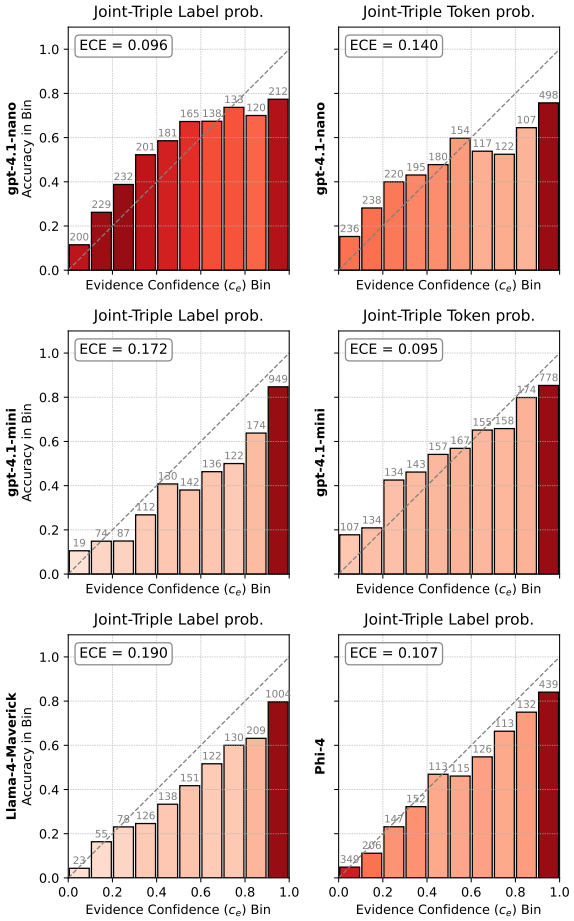

Figure 2: Reliability diagrams for evidence confidence calibration comparing Label prob. and Token prob. (where available). Label prob. demonstrates consistent near-diagonal calibration across all models (ECE 0.096-0.190), while Token prob. shows model-dependent performance (GPT-4.1-nano: 0.140; GPT-4.1-mini: 0.095).

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.[9]

Table 2 shows spurious correctness detection performance using Label prob., our best-performing confidence method (§5.1). Evidence-level confidence consistently outperforms both chain-level (Δ+0.13 to +0.30 ROC-AUC) and answer-level (Δ+0.04 to +0.19) across all models, with Phi-4 achieving the highest ROC-AUC of 0.84.

The chain-level baseline assigns one verbalized confidence score to the entire reasoning

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

Model	Ans	Chain	Ev
GPT-4.1-mini	0.59	0.62	**0.74**
Llama-4-Maverick	0.53	0.52	**0.69**
Phi-4	0.65	0.54	**0.84**

Table 2: Spurious correctness detection (ROC-AUC). Ans: answer-level, Chain: chain-level baseline (verbalized), Evi: evidence-level. All use Label prob. except Chain. Evidence-level consistently outperforms both coarser granularities.

chain (§4.2), while answer- and evidence-level use Label prob. (§5.1). The substantial evidence-over-chain improvements confirm that fine-grained triplet-level confidence is necessary —coarse-grained chain-level confidence is insufficient even when intermediate reasoning is available. To verify robustness across confidence extraction methods, we additionally evaluated all three granularities using verbalized confidence, confirming that evidence-level consistently outperforms answer-level across all models (Appendix B.1.3).

Cross-lingual evaluation on 2WikiMultiHopQA (English) replicates these findings, with evidence-level showing substantial improvements over chain-level (Δ+0.14 to +0.32) and answer-level (Δ+0.10 to +0.26), demonstrating that fine-grained confidence addresses fundamental multi-hop reasoning challenges across languages (detailed results in Appendix B.1.1).

These consistent improvements across models, methods, and languages 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. The quantitative effectiveness of this approach is further demonstrated through ROC and PR curves in Appendix Fig. 5.

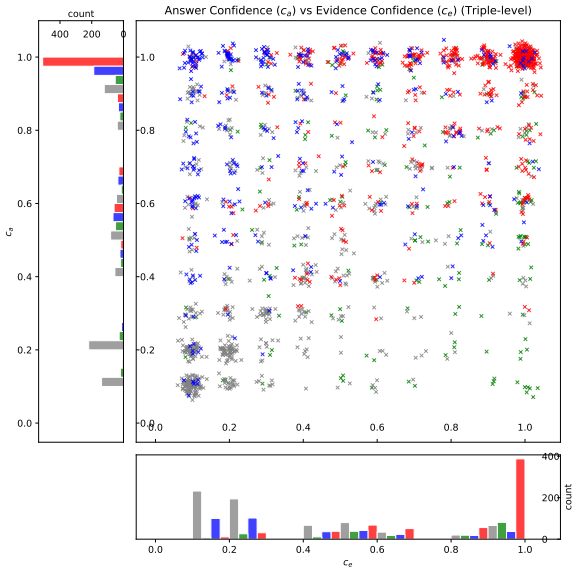

Figure 3: Answer confidence vs evidence confidence scatter plot (Label prob.). **Red**: Both answer and evidence correct (true correct), **Blue**: Answer correct but evidence wrong (spurious correctness), **Green**: Evidence correct but answer wrong, **Gray**: Both answer and evidence wrong. The histograms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

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?

Our results confirm this hypothesis. Table 3[10] on JEMHopQA shows that joint generation of answer and evidence confidence 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 improvements range from 12% to 32% across three models, demonstrating better discrim-

[10]Only-answer and joint generation prompts are provided in Appendix Tables 6 and 5.

Model	Method	Only-Answer ECE/AUC	Joint-answer ECE/AUC	Improv. Rate ECE/AUC
GPT-4.1-mini	Label prob.	0.23/0.72	0.17/0.84	26%/18%
Llama-4-Maverick	Verb. 1S	0.42/0.58	0.24/0.77	**43%/32%**
Phi-4	Label prob.	0.14/0.67	0.16/0.75	-16%/12%

Table 3: Answer confidence performance on JEMHopQA: 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.

ination. Phi-4's ECE worsened (-16%), which may reflect its already-low baseline ECE (0.14) leaving less room for improvement. However, the 12% 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 C.1) 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 C.2), 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 on JEMHopQA, 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 14 for details):

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 con-

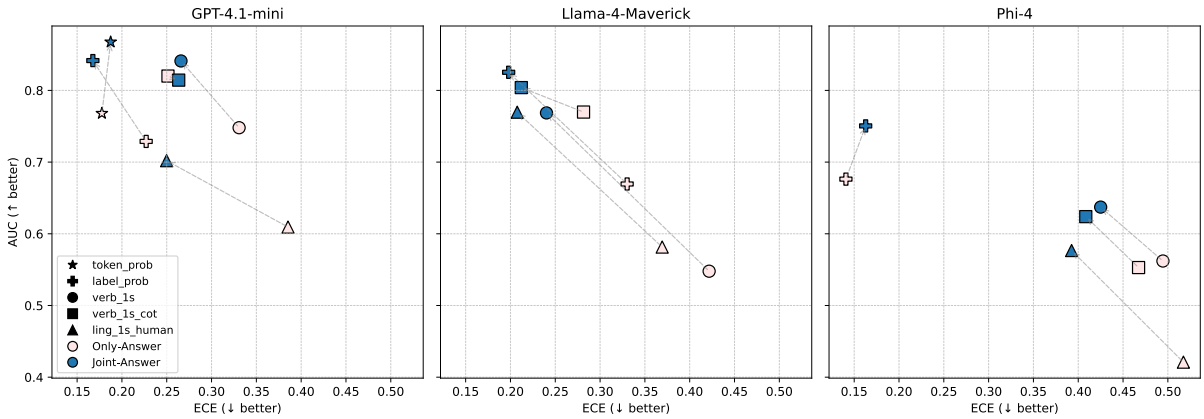

Figure 4: Plot of answer confidence for the baseline Answer-only method versus the Joint-Answer method (simultaneous evidence generation) across all models.

fidence often reflects legitimate uncertainty (detailed breakdown in Appendix Table 15):

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; practical recommendations for deployment are provided in Appendix D.3.

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: Our framework requires decomposable evidence units with alignable gold annotations for automatic evaluation. We instantiate this as (Subject, Relation, Object) triplets for JEMHopQA and 2WikiMultiHopQA, which work well for factual multi-hop QA but may not suit all reasoning types. However, the core principle—assigning confidence to decomposable reasoning components—can be adapted to other formats such as table rows, slot-value facts, or knowledge graph paths, as long as each component has clear correctness criteria and gold units allow one-to-one matching after normalization. Future work should explore such extensions and investigate the trade-offs between structural expressiveness and evaluation scalability.

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.

Potential risks: Over-reliance on confidence scores without validation could lead to misplaced trust, particularly under distribution shift or in cross-lingual settings beyond our evaluation. We recommend human oversight in high-stakes applications and domain-specific validation before deployment.

Ethical considerations

We follow the ACL Publication Ethics and the ARR Responsible NLP Research checklist. We used ChatGPT 5 and Claude Sonnet 4.5 only as assistive tools (code refactoring and copy-editing); all content was author-verified and the models are not authors. No new human-subject data were collected and no PII or confidential information were used; IRB approval was not required. We evaluate on public datasets (e.g., JEMHopQA; 2WikiMultiHopQA) and access models via official APIs under their terms; we do not redistribute third-party artifacts. Prompts and evaluation code will be released for reproducibility.

Acknowledgments

This work was supported by BIPROGY Inc., which provided the computing environment and research funding. Part of the initial study was conducted while the author was working at RIKEN.

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.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Caiqi Zhang, Ruihan Yang, Zhisong Zhang, Xinting Huang, Sen Yang, Dong Yu, and Nigel Collier. 2024. Atomic calibration of LLMs in long-form generations. *Preprint*, arXiv:2410.13246. ArXiv:2410.13246.

A Detailed Experimental Setup

A.1 Implementation Details and Reproducibility

Datasets and splits. We evaluate on 1,000 items from JEMHopQA training split and 300 items from 2WikiMultiHopQA dev split. We use k=3 few-shot exemplars per dataset, selected to have distinct relations and single-entity answers. Exemplar IDs are fixed and documented in the code repository. We do not redistribute third-party datasets.

Models and endpoints. All experiments use API inference only. Model versions: GPT-4.1-mini (2025-04-14), Llama-4-Maverick-17B-128E-Instruct-FP8 (v1, Oct 2024), Phi-4 (v7, Oct 2024). Code ran on Linux with Python 3.11.

Prompting and decoding. Label prob. uses temperature $= 0.7$, top-p $= 0.95$, n=10 samples. All other methods use temperature $= 0.0$ (greedy decoding). Max output tokens: 2,048. No custom stop sequences. All prompts use the same three few-shot exemplars per dataset (fixed exemplar IDs in repository).

Normalization. We apply a single normalization pipeline to all content strings (answers and the subject/relation/object fields of triples) *after* parsing the "(subject, relation, object)" structure:

1. Unicode normalization (NFKC; includes full/half-width unification)
2. Trim leading/trailing whitespace
3. Map Japanese "はい/いいえ" or "yes/no" prefixes to YES/NO
4. Convert Japanese numerals to Arabic (JP data only)
5. Remove non-essential bracketed annotations (e.g., "(年)", "(entity)"), not affecting structural delimiters
6. Remove Japanese quotation marks (「」『』)
7. Strip non-structural punctuation (triple delimiters are preserved for parsing)
8. Collapse internal whitespace
9. Lowercase ASCII letters

For numbers: drop thousands separators. For dates: unify Japanese era ↔ Western year mappings when applicable. Triples are first parsed by splitting on the two outer commas while ignoring commas inside parentheses; the above normalization is then applied to each field. Normalization code is available in the repository.

Triple parsing. Triples are parsed from the $(\text{subject}, \text{relation}, \text{object})$ format by splitting on the two outer commas (commas inside fields/parentheses are preserved). After parsing, normalization is applied to each field. During aggregation, the order of triples within a prediction is ignored and duplicates are preserved (multiset); the field order within each triple is preserved.

Answer and evidence aggregation for Label prob. Across n=10 samples, we compute empirical frequencies after normalization.

Answer aggregation: The final answer is the majority string (exact match after normalization). Its confidence equals the majority frequency divided by n. Ties are broken by first occurrence (deterministic under a fixed seed).

Evidence aggregation: Each trajectory yields a (multi)set of normalized triples. We canonicalize each set by sorting triples lexicographically (subject → relation → object) and use the sorted tuple as a key, which ignores within-trajectory order while preserving duplicates. The most frequent evidence set E^* is selected as the final evidence (ties broken by first occurrence).

For each triple $e \in E^*$, the evidence confidence is

$$p(e \mid q) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}[\, e \in E^{(i)} \,],$$

i.e., the fraction of trajectories containing e.

Evidence evaluation against gold. We evaluate predicted triples against gold annotations using GPT-4.1 (2025-04-14; temperature=0.0, $n = 1$, max_tokens=4096) via constrained function calling.

Evaluation protocol: The system prompt enforces calling evaluate_triples and the fixed output order "pred $1..m$ → gold $1..k$". The model returns a JSON results array with one entry per predicted and per gold triple, each including type ("pred"/"gold"), index (1-based), triple, matched_index (or null), and a binary $\text{score} \in \{0.0, 1.0\}$.

Exact sequence match: Before invoking the LLM, if the *entire sequence* of predicted triples exactly matches the gold sequence after normalizing parenthesis glyphs (全角 () ／ ASCII ()), we synthesize per-entry outputs without an API call by assigning score=1.0 to all predicted and gold entries and setting matched_index=1..m accordingly.

Scoring rubric: Summarized in Table 4 (e.g., synonyms/abbreviations and unit/granularity equivalence count as correct; format violations and factual errors are incorrect).

Metrics and calibration. We report Expected Calibration Error (ECE; 10 equal-width bins), Brier Score (BS), and Selective AUC (area under the accuracy–coverage curve). For discrimination, we also report ROC-AUC and PR-AUC for evidence-level spurious-correctness detection.

Temperature scaling is evaluated with 5-fold cross-validation (KFold with shuffling; seed 42).

In each fold, a temperature is fitted on the training split and ECE is computed on the held-out split; we report the mean ECE across folds together with the mean and standard deviation of the fitted temperatures.

Randomness and retries. Cross-validation uses KFold with shuffling (seed=42). For Label prob. generation we use temperature=0.7, top-p=0.95, and n=10; the API does not expose a stable random seed, so small run-to-run variation is possible, and all reported numbers come from a single run. Evidence evaluation via function calling uses temperature=0.0 and n=1 (effectively deterministic modulo API behavior). All API calls include retry logic (up to 3 retries); items that still fail are excluded from analysis.

Code and data release. We release: (1) all prompt templates (Japanese and English); (2) evaluation scripts for automated judgment and metric computation; (3) few-shot exemplars embedded as IDs within the prompt templates; (4) normalization and triple-parsing utilities; and (5) runnable configuration (CLI arguments and sample scripts) specifying model identifiers/versions and decoding parameters (the model version can be passed as an argument). We do not redistribute third-party datasets or model weights; users should obtain them from the official sources cited in the paper. Access to commercial APIs (e.g., GPT-4.1 / GPT-4.1-mini) must be obtained via the providers'standard procedures; API keys and private endpoints are not included in our release. The public repository is available at https://github.com/aiishii/finegrained_conf and includes the LICENSE file.

A.2 Automated Evaluation

All evaluation metrics require binary correctness labels for each answer and evidence triplet. We obtain these labels using GPT-4.1 with a constrained function-calling interface (details in §A.1). Table 4 presents the prompt template used for evidence evaluation with one-to-one matching and binary scoring.

Answer evaluation. We use exact match for YES/NO questions (33% of JEMHopQA). For entity-based questions (67%), we employ GPT-4.1 to judge semantic equivalence when exact match fails.

Evidence evaluation. The model performs one-to-one matching between predicted and gold triples, assigning binary scores (1.0 or 0.0) based on the rubric in Table 4. The rubric tolerates surface-form variation (synonyms, abbreviations, subject-object swaps for symmetric relations) while requiring semantic equivalence.

Reliability assessment. To validate the automated evaluation, one author manually labeled 100 randomly sampled instances per model (300 total). Agreement rates: answer correctness 98% (GPT-4.1-mini), 100% (Llama-4-Maverick), 98% (Phi-4); evidence correctness 93%, 94%, 95% respectively. While these error rates are non-negligible, they affect all methods equally, preserving the validity of relative comparisons.

A.3 Prompt Templates

This section presents the prompts used in our experiments. As our evaluation was conducted on the Japanese JEMHopQA dataset, all prompts were originally written in Japanese. We provide English translations for clarity, followed by Japanese examples in §A.3.2.

A.3.1 English Translation

The following tables present the prompt templates used in our experiments. Table 5 shows the prompts for our main joint generation approach, while Table 6 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.

A.3.2 Japanese Prompt Examples

For reference, we provide examples of the actual Japanese prompts used in our experiments. These correspond to the English translations of the Label prob. prompt in §A.3.1.

B Additional Experimental Results

B.1 Comprehensive Evidence Confidence Results

Table 8 presents complete results for evidence confidence across all methods, models, and datasets.

Component	Prompt Template
System	You evaluate evidence triples for multi-hop QA. Given lists of gold and predicted triples, first establish an optimal one-to-one alignment that maximizes overall semantic equivalence, then return a JSON results array in which each item represents either a predicted or a gold triple, including fields type (“pred” or “gold”), index (1-based), triple (“(subject, relation, object)”), matched_index (or null), and score (either 1.0 or 0.0). Scoring is binary per matched pair.
User	Gold triples (1-based): 1. ({GOLD_SUBJECT_1}, {GOLD_RELATION_1}, {GOLD_OBJECT_1}) 2. ({GOLD_SUBJECT_2}, {GOLD_RELATION_2}, {GOLD_OBJECT_2}) ... Predicted triples (1-based): 1. ({GEN_SUBJECT_1}, {GEN_RELATION_1}, {GEN_OBJECT_1}) 2. ({GEN_SUBJECT_2}, {GEN_RELATION_2}, {GEN_OBJECT_2}) ... Judging rules for score = 1.0 (equivalence allowed): • Synonyms / spelling variants / abbreviations; entity surface-form differences • Information equivalence sufficient to answer the question (unit/granularity differences acceptable) • Subject↔Object swap for symmetric relations (e.g., spouse) • Presuppositions and necessary inference elements for reasoning chains • Boundary values of ranges; components of composite (AND) conditions Cases for score = 0.0: • Missing/misidentified core information; only a fragment that does not contribute to derivation • Factually incorrect statements (beyond conventional/format variation) • Improper format (no entity/value; boolean or free-text sentence), vague terms (“around”, “many”) • Irrelevant to the question or to the aligned gold triple; missing elements in a composite chain Output JSON schema (abbreviated): {“results”: [{“type”:“pred”,“index”:1, “triple”:“(S,R,O)”, “matched_index”:2, “score”:1.0}, {“type”:“gold”,“index”:2, “triple”:“(S,R,O)”, “matched_index”:1, “score”:1.0}, ...]}

Table 4: Prompt template for automated evidence evaluation with one-to-one alignment and binary scoring. {GOLD_*} and {GEN_*} placeholders are replaced with actual components. The complete JSON schema is provided in the appendix.

Accuracy indicates the proportion of correctly generated evidence triplets, while other metrics evaluate calibration (ECE, BS) and discrimination (AUC, ROC, PR).

B.1.1 Cross-lingual Validation Results

To validate the generalizability of our approach across languages, we evaluated on 2WikiMultiHopQA (English, 300 dev samples) in addition to JEMHopQA (Japanese, 1,000 samples). Complete results for both datasets, including all calibration and discrimination metrics, are presented in Table 8 (§B.1). This section focuses on cross-lingual comparisons and robustness analysis.

B.1.2 Granularity Effects Across Languages

Table 9 shows spurious correctness detection performance (ROC-AUC, Label prob.) across all granularity levels for both datasets.

Key observations. (1) **Evidence-level superiority over chain-level**: Evidence-level consistently and substantially outperforms chain-level across all six combinations (Δ+.129 to +.316 ROC-AUC, mean +.225). This demonstrates that fine-grained triplet-level confidence is necessary even when chain-level reasoning is available—coarse-grained chain-level confidence is insufficient for effective spurious correctness detection. (2) **Evidence-level superiority over answer-level**: Evidence-level also consistently outperforms answer-level across all combinations (Δ+.099 to +.261 ROC-AUC, mean +.176), confirming that structured intermediate reasoning confidence provides substantial benefits over final-answer confidence alone. (3) **Chain-level baseline variability**: Chain-level shows mixed performance relative to answer-level (higher in 2/6 cases), reflecting the challenges of obtaining reliable verbalized confidence for variable-length reasoning chains. (4) **Cross-lingual robustness**: The evidence-level advantages persist across Japanese and English datasets and across diverse model architectures (GPT-4 variants, Llama-4-Maverick), indicating that the benefits of fine-grained confidence are fundamen-

Method	Template (Joint)
Baseline (Chain)	Provide the answer and the supporting triples that lead to your conclusion. First, show your reasoning process step by step, then output the supporting triples and the final answer. Triples must be in the form (Subject, Relation, Object). The subject must be an entity, and the object must be either an entity or a specific value (date, number, etc.). Use short single phrases for all fields. Finally, report your confidence (0.00–1.00, two decimals) in your overall reasoning process (Thought + Triples) and, separately, your confidence in the answer. Output in the following format: Thought: [reasoning process] Triple 1: (Subject, Relation, Object) Triple 2: (Subject, Relation, Object) ... Overall reasoning confidence: 0.00–1.00 Answer: YES|NO|<short single phrase> 0.00–1.00 [Examples omitted] Question: {THE_QUESTION}
Label prob. / Token prob.	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> [Examples omitted] Question: {THE_QUESTION}
Verb. 1S	(same as 1st row of Label prob.) 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 [Examples omitted] Question: {THE_QUESTION}
Verb. 1S CoT	(same as 1st row of Label prob.) 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 [Examples omitted] Question: {THE_QUESTION}
Ling. 1S	(same as 1st row of Label prob.) 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 [Examples omitted] Question: {THE_QUESTION}

Table 5: 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 6: 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.

tal rather than language- or model-specific.

B.1.3 Robustness Across Confidence Methods

To confirm that granularity effects are method-independent, Table 10 shows results using verbalized confidence across all granularities on both datasets.

B.1.4 Cross-lingual Summary

The cross-lingual evaluation confirms: (1) **Monotonic improvement**: Evidence-level consistently outperforms answer-level across both Japanese and English for all evaluated models. (2) **Robustness across model families**: Benefits persist across GPT-4 variants, Llama-4, and Phi-4. (3) **Calibration improvements replicate**: Label prob. achieves ECE of 0.111-0.165 on 2WikiMultiHopQA compared to 0.173-0.191 on JEMHopQA (Table 8). (4) **Method independence**: Granularity effects persist across both sampling-based (Label prob.) and verbalized confidence methods (Table 10). These results validate that our structured triplet-based approach addresses fundamental challenges that transcend language boundaries.

B.2 Model Size Effects

Table 11 compares different sizes within the GPT-4.1 family. Both nano and mini show similar improvement patterns with Label prob., suggesting robustness across model scales.

C Ablation Studies

C.1 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:

- **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 12 reveals two key findings:

Evidence generation improves accuracy: Comparing C1 to C2, we observe substantial

Method	Template (Joint)
Label prob. / Token prob.	次の質問に対する回答と、その結論に至った根拠となるトリプルを提供してください。 トリプルは (主語, 関係, 目的語) の形式。 主語はエンティティ、目的語はエンティティまたは具体的な値（日付、数値等）とし、いずれも短い単一フレーズとする。 以下の形式で出力してください： トリプル 1: (主語, 関係, 目的語) トリプル 2: (主語, 関係, 目的語) ... 回答: YES|NO|< 短い単一フレーズ > 例： 質問: Wii U とニンテンドーゲームキューブ、日本での本体と同時発売のゲームソフト数が多いのはどちらでしょう？ トリプル 1: (Wii U，日本での本体と同時発売のソフト数，11 タイトル) トリプル 2: (ニンテンドーゲームキューブ，日本での本体と同時発売のソフト数，2 本) 回答: Wii U [Two examples omitted] 質問: {THE_QUESTION}
Verb. 1S	(same as the first three rows of Label prob.) 各トリプルと回答の正確さに対する確信度（0.00–1.00, 小数 2 桁）も示すこと。 以下の形式で出力してください： トリプル 1: (主語, 関係, 目的語) 0.00–1.00 トリプル 2: (主語, 関係, 目的語) 0.00–1.00 ... 回答: YES|NO|< 短い単一フレーズ > 0.00–1.00 例： 質問: Wii U とニンテンドーゲームキューブ、日本での本体と同時発売のゲームソフト数が多いのはどちらでしょう？ トリプル 1: (Wii U，日本での本体と同時発売のソフト数，11 タイトル) [確信度] トリプル 2: (ニンテンドーゲームキューブ，日本での本体と同時発売のソフト数，2 本) [確信度] 回答: Wii U [確信度] [Two examples omitted] 回答: YES [確信度] 質問: {THE_QUESTION}

Table 7: Example Japanese prompt (Label prob. / Token prob. and Verb. 1S method). All methods require evidence generation before the final answer. {THE_QUESTION} is replaced with the actual question.

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.

C.2 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 13 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.

Note: These preliminary experiments used a smaller dataset and slightly different evaluation criteria than the main experiments, hence the abso-

Dataset	Model	Method	Acc↑	ECE↓	ECE-t↓	BS↓	BS-t↓	AUC↑	ROC↑	PR↑
JEMHopQA (Japanese)	GPT-4.1 -nano	Baseline	.569	.313	**.044**	.325	.238	**.683**	.576	.420
		Label prob.	.516	**.096**	.128	**.217**	.233	.677	**.751**	**.673**
		Token prob.	.496	.140	.065	.235	**.216**	.666	.521	.407
		Verb. 1S	.500	.322	.109	.327	.237	.667	.723	.717
		Verb. 1S CoT	.540	.346	.114	.346	.241	.679	.672	.624
		Ling. 1S	.542	.294	.049	.330	.246	.536	.549	.544
	GPT-4.1 -mini	Baseline	.626	.295	.064	.307	.225	.728	.615	.528
		Label prob.	.621	.172	.139	.221	.197	.781	**.744**	.565
		Token prob.	.645	**.095**	.072	**.192**	**.186**	**.816**	.537	.292
		Verb. 1S	.629	.297	.125	.300	.210	.791	.733	.598
		Verb. 1S CoT	.628	.305	.140	.312	.223	.754	.707	.576
		Ling. 1S	.615	.288	**.054**	.307	.227	.669	.673	**.670**
	Llama-4 -Maverick	Baseline	.645	.285	.093	.299	.232	.712	.524	.386
		Label prob.	.611	**.190**	.145	**.245**	**.218**	**.733**	**.693**	.552
		Verb. 1S	.602	.316	.137	.317	.243	.710	.679	.613
		Verb. 1S CoT	.618	.295	.086	.302	.237	.716	.685	.614
		Ling. 1S	.589	.297	**.079**	.309	.230	.652	.659	**.617**
	Phi-4	Baseline	.458	.516	.337	.507	.348	.545	.535	.488
		Label prob.	.449	**.107**	.188	**.178**	**.204**	**.695**	**.839**	**.825**
		Verb. 1S	.459	.508	.326	.495	.329	.574	.622	.584
		Verb. 1S CoT	.463	.500	.297	.491	.327	.549	.647	.587
		Ling. 1S	.447	.491	**.124**	.486	.261	.457	.549	.606
2WikiMHQA (English)	GPT-4.1	Baseline	.711	.245	**.047**	.259	.199	.779	.508	.157
		Label prob.	.717	**.108**	.122	**.145**	**.154**	.844	**.824**	**.689**
		Token prob.	.674	.156	.106	.210	.192	.801	.513	.337
		Verb. 1S	.746	.212	.064	.226	.178	**.860**	.691	.295
	GPT-4.1 -mini	Baseline	.510	.427	.060	.428	.253	.496	.478	.174
		Label prob.	.511	**.163**	.158	**.204**	**.206**	**.704**	**.771**	**.703**
		Token prob.	.481	.232	.122	.282	.239	.607	.578	.487
		Verb. 1S	.506	.405	**.076**	.402	.246	.600	.645	.544
	Llama-4 -Maverick	Baseline	.455	.493	.218	.483	.291	.535	.548	.494
		Label prob.	.395	**.140**	**.209**	**.194**	**.220**	**.605**	**.730**	**.720**
		Verb. 1S	.319	.564	.277	.543	.320	.568	.677	.636

Table 8: Comprehensive evidence confidence results across all methods, models, and datasets. Acc: evidence triplet accuracy; ECE/BS: calibration metrics; AUC/ROC/PR: discrimination metrics (ROC and PR for spurious correctness detection at evidence level). Bold indicates best performance per model-dataset combination. For GPT-4.1 variants on JEMHopQA, Token prob. achieves the lowest calibration error for mini (ECE .095) but shows model-dependent behavior with higher error for nano (.140 vs Label's .096). Label prob. demonstrates consistent performance across all models (ECE .096-.190) and superior spurious correctness detection (highest ROC/PR across all models). Note that high accuracy does not guarantee good spurious correctness detection (ROC/PR), highlighting the importance of fine-grained confidence estimation.

Dataset	Model	Answer -level	Chain -level	Evidence -level	Δ Evi-Chain	Δ Evi-Ans
JEMHopQA (Japanese)	GPT-4.1-mini	.594	.615	.744	+.129	+.150
	Llama-4-Maverick	.533	.524	.693	+.169	+.160
	Phi-4	.651	.535	.839	+.304	+.188
2WikiMultiHopQA (English)	GPT-4.1	.563	.508	.824	+.316	+.261
	GPT-4.1-mini	.572	.478	.771	+.292	+.198
	Llama-4-Maverick	.630	.592	.730	+.138	+.099

Table 9: Cross-lingual comparison of spurious correctness detection (ROC-AUC, Label prob.). Evidence-level confidence consistently and substantially outperforms both chain-level (Δ+.129 to +.316) and answer-level (Δ+.099 to +.261) across all six model-dataset combinations. Chain-level baseline uses verbalized confidence and shows variable performance relative to answer-level.

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 12: 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.

Dataset	Model	Ans	Chain	Evi	Δ(E-A)
JEMHop (JP)	GPT-4.1-mini	0.664	0.615	**0.733**	+0.069
	GPT-4.1-nano	0.617	0.576	**0.723**	+0.106
	Llama-4-Mav.	0.640	0.524	**0.679**	+0.039
	Phi-4	0.613	0.535	**0.622**	+0.008
2Wiki (EN)	GPT-4.1	0.553	0.508	**0.691**	+0.138
	GPT-4.1-mini	0.588	0.478	**0.645**	+0.058
	Llama-4-Mav.	0.538	0.592	**0.683**	+0.144

Table 10: Spurious correctness detection (ROC-AUC) using verbalized confidence (Verb. 1S) at all granularities. Evidence-level consistently outperforms answer- and chain-level across all seven combinations, confirming that the benefit of fine-grained confidence is robust across confidence extraction methods.

Dataset	Model	Δ ROC (E-A)	ECE-t (Label prob.)
JEMHop (JP)	GPT-4.1-mini	+0.150	0.139
	GPT-4.1-nano	+0.156	0.128
2Wiki (EN)	GPT-4.1	+0.261	0.122
	GPT-4.1-mini	+0.198	0.158

Table 11: Model size comparison within GPT-4.1 family. Δ ROC (E-A): Evidence-Answer improvement. Similar gains across sizes suggest methodological rather than capacity-driven improvements.

Method	Answer Confidence			Evidence Confidence		
	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 13: 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.

lute numbers differ from those reported in the main text.

D Analysis and Application

D.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.

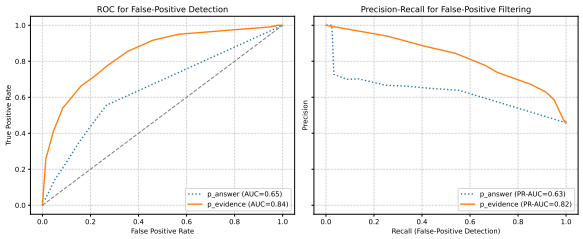

Figure 5: ROC and PR curves for spurious correctness detection using Phi-4/Label prob. . Evidence confidence (orange) achieves ROC-AUC 0.84 and PR-AUC 0.82, significantly outperforming answer confidence (blue) with ROC-AUC 0.65 and PR-AUC 0.63.

D.2 Detailed Error Analysis Tables

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

D.3 Practical Deployment Considerations

Based on our empirical findings across multiple models and two languages, we offer the following observations that may inform deployment. Results and ranges below refer to our JEMHopQA and 2WikiMultiHopQA settings unless noted.

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 14: 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 15: Distribution of patterns in low-confidence correct evidence (n=90, 30 samples per model).

Method selection: Label prob. showed comparatively strong calibration (ECE 0.096−0.190) across all tested architectures. While it requires multiple samples (n=10 in our runs), the calibration gains can justify the added latency where confidence reliability is critical.

Granularity choice: Evidence-level confidence yielded higher spurious correctness detection in our experiments (ROC-AUC 0.69−0.84) than answer- or chain-level alternatives. For deployment, we recommend tuning decision thresholds on a held-out dev set from the target domain.

Error monitoring: We observed that about 49% of high-confidence errors in our sample involved numerical or temporal fields. Prioritizing verification on these field types may improve overall reliability.

Calibration maintenance: Monitor ECE (or a related calibration metric) on production data over time; distribution shifts may necessitate periodic temperature recalibration.

While these observations are derived from multi-hop QA, we expect the general idea of fine-grained confidence to potentially extend to other reasoning-heavy tasks where intermediate steps can be decomposed into verifiable units. Practitioners should validate these patterns in their specific application contexts.