Double-Calibration: Towards Reliable LLMs via Calibrating Knowledge and Reasoning Confidence

Yuyin Lu[1] , **Ziran Liang**[2] , **Yanghui Rao**[1*] , **Wenqi Fan**[2] , **Fu Lee Wang**[3] and **Qing Li**[2]
[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[2]Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
[3]School of Science and Technology, Hong Kong Metropolitan University, Hong Kong SAR
luyy37@mail2.sysu.edu.cn, raoyangh@mail.sysu.edu.cn

Abstract

Reliable reasoning in Large Language Models (LLMs) is challenged by their propensity for hallucination. While augmenting LLMs with Knowledge Graphs (KGs) improves factual accuracy, existing KG-augmented methods fail to quantify epistemic uncertainty in both the retrieved evidence and LLMs' reasoning. To bridge this gap, we introduce DoublyCal, a framework built on a novel double-calibration principle. DoublyCal employs a lightweight proxy model to first generate KG evidence alongside a calibrated evidence confidence. This calibrated supporting evidence then guides a black-box LLM, yielding final predictions that are not only more accurate but also well-calibrated, with confidence scores traceable to the uncertainty of the supporting evidence. Experiments on knowledge-intensive benchmarks show that DoublyCal significantly improves both the accuracy and confidence calibration of black-box LLMs while maintaining low token cost.

1 Introduction

The reliability of Large Language Models (LLMs) is critically undermined by their tendency to hallucinate [Huang *et al.*, 2024], a problem rooted in both intrinsic *epistemic uncertainty* (knowledge gaps) and extrinsic *aleatoric uncertainty* (data ambiguity) [Hüllermeier and Waegeman, 2021]. To mitigate this, Knowledge Graph-augmented Retrieval-Augmented Generation (KG-RAG) has emerged as a leading paradigm [Zhang *et al.*, 2025a]. By augmenting LLM with structured evidence retrieved from external Knowledge Graphs (KGs), KG-RAG is helpful to reduce the model's internal knowledge gaps and improve the factual accuracy of its responses [Xiang *et al.*, 2025].

However, this KG-augmentation mechanism introduces a new yet critical dependency: *the certainty of the retrieved evidence itself*. Prevailing KG-RAG methods [Luo *et al.*, 2024; Li *et al.*, 2025a; Liu *et al.*, 2026] often rely on an idealistic assumption that the retrieved evidence is always both sufficient and certain to support correct reasoning for a given query. This assumption is routinely violated in practice due to ambiguous queries, the intrinsic incompleteness of KGs, and imperfections in the retrieval process. Consequently, when provided with partial evidence, LLMs may still produce confidently stated but incorrect predictions [Kalai *et al.*, 2025]. For example, as illustrated in Figure 1, given the partial evidence "Belle is a sibling of Snoopy", an LLM might incorrectly infer "Belle is Snoopy's brother". Thus, current KG-RAG lacks the ability to assess and control uncertainty at the very source of the reasoning chain.

Concurrently, research on Uncertainty Quantification (UQ) for LLMs aims to calibrate prediction confidence but focuses predominantly on the final output [Xia *et al.*, 2025]. For instance, verbalized UQ methods elicit confidence estimates from black-box LLMs, yet these estimates remain opaque and non-traceable [Tian *et al.*, 2023; Xiong *et al.*, 2024]. It is impossible to discern whether the expressed uncertainty stems from flawed evidence, deficiencies in the model's own reasoning, or the intrinsic difficulty of the task. Therefore, existing UQ methods cannot synergize effectively with KG-RAG to provide a stepwise-calibrated view of the complete evidence-to-prediction chain.

In summary, a principled solution for systematically managing the propagation of uncertainty in KG-augmented LLMs is lacking. To bridge this gap, we propose a novel **double-calibration** paradigm. Its core lies in moving beyond basic evidence retrieval to the construction of *a calibrated reasoning chain*, where confidence is explicitly estimated and made traceable from the retrieved KG evidence to the final LLM prediction. As shown in Figure 1, this enables the LLM to weigh alternative answers (e.g., correctly favoring "Spike" over "Belle") based on the calibrated confidence of the supporting evidence, rather than making an overconfident guess.

We instantiate this principle in **DoublyCal**, a framework that implements double-calibrated KG-RAG. DoublyCal grounds the LLM's reasoning on verifiable KG evidence and performs dual calibration: it first calibrates the confidence of the retrieved evidence, then uses this calibrated evidence to guide and further calibrate the final LLM prediction. Specifically, we formalize KG evidence as constrained relational paths extracted from a KG. We then train a lightweight

*Corresponding Author.

This work is to appear in the Proceedings of the 35th International Joint Conference on Artificial Intelligence (IJCAI 2026). A direct link to the conference version will be released at that time.

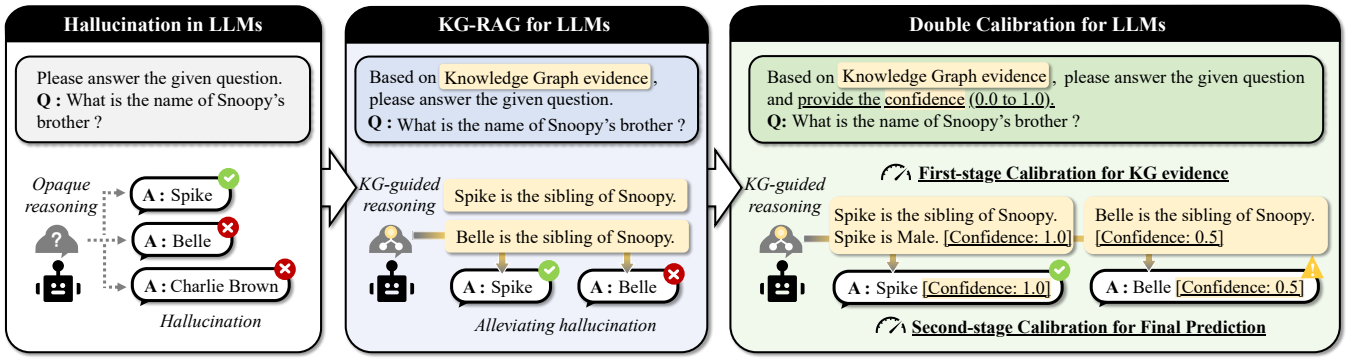

Figure 1: A motivating example of double-calibration against hallucination in KG-augmented LLMs.

proxy model under Bayesian supervision to generate relevant KG evidence alongside a calibrated confidence score for each query. During inference, the primary LLM is prompted with both the KG evidence and its confidence estimate, leading to more accurate and better-calibrated predictions. Crucially, because the evidence confidence explicitly estimates the expected reasoning uncertainty of the LLM when utilizing the provided evidence, the final confidence becomes traceable to the verifiable KG evidence and its calibrated confidence, rather than remaining an opaque global estimate. Our main contributions are summarized as follows:

- We establish the principle of double-calibration for reliable KG-augmented LLMs, which mandates explicit confidence calibration for both the KG evidence and the final LLM predictions.
- We propose DoublyCal[1], a framework that implements this principle via a Bayesian-calibrated proxy model, providing the primary LLM reasoner with KG evidence accompanied by evidence confidence.
- We empirically demonstrate that DoublyCal significantly and consistently improves the accuracy and calibration of diverse black-box LLMs on knowledge-intensive benchmarks in a cost-effective manner.

2 Related Work

2.1 Knowledge-Augmented Generation for Reliable LLMs

Retrieval-Augmented Generation (RAG) reduces the inherent knowledge gaps of LLMs by providing external information, thereby improving the factual accuracy of their responses [Zhang *et al.*, 2025a]. The choice of knowledge source defines a spectrum of RAG variants, ranging from (i) unstructured text in Vanilla RAG [Guo *et al.*, 2025; Sun *et al.*, 2025], to (ii) textual graphs that model latent connections in GraphRAG [He *et al.*, 2024; Li *et al.*, 2025b], and finally to (iii) formal Knowledge Graphs (KGs) with explicit relations in KG-RAG [Luo *et al.*, 2024; Li *et al.*, 2025a]. By providing precise and structured knowledge, KG-RAG offers a rigorous foundation for complex reasoning and has demonstrated superior performance on knowledge-intensive tasks [Xiang *et al.*, 2025; Pan *et al.*, 2024].

However, the retrieved knowledge itself may be noisy or insufficient, posing a persistent challenge to the reliability of LLM reasoning. To address this, some research dynamically selects external knowledge when it conflicts with the LLM's internal parametric knowledge [Liu *et al.*, 2026; Zhang *et al.*, 2025b]. Unlike prior work that focus on resolving knowledge conflicts, we introduce a double-calibration principle which explicitly quantifies confidence for both the retrieved evidence and the final LLM prediction, thereby systematically identifying epistemic boundaries.

2.2 Uncertainty Quantification for LLMs

Uncertainty Quantification (UQ) for LLMs aims to calibrate the confidence of model predictions to identify their epistemic boundaries [Huang *et al.*, 2024]. While some studies incorporate uncertainty awareness during training [Stangel *et al.*, 2025], most practical UQ methods often operate post-hoc and are categorized by model access [Xia *et al.*, 2025].

For open-source LLMs, confidence is typically derived from internal states, such as the feature-space distribution of hidden embeddings [Chen *et al.*, 2024; Vazhentsev *et al.*, 2025] and the predictive entropy of the output distribution [Malinin and Gales, 2021]. For black-box LLMs, methods rely on API-based probing. A prevalent strategy generates multiple responses and evaluates their semantic consistency [Manakul *et al.*, 2023; Kuhn *et al.*, 2023; Lin *et al.*, 2024]. A more efficient alternative is verbalized UQ, which directly prompts the LLM to verbalize its own confidence, eliciting introspective uncertainty estimates [Tian *et al.*, 2023; Xiong *et al.*, 2024; Tanneru *et al.*, 2024]. Its plug-and-play nature and low cost make verbalized UQ readily integrable with KG-RAG, forming a strong single-calibration baseline that calibrates only the final output.

A fundamental limitation across prior UQ paradigms is their exclusive focus on the final output, deriving confidence solely from the LLM's internal states or self-assessment. This makes them vulnerable to the model's overconfidence biases due to the lack of an objective external anchor [Xiong *et al.*, 2024]. Our double-calibration principle addresses this by first calibrating externally verifiable KG evidence, thereby establishing a traceable foundation for the entire reasoning chain.

[1]The source code of DoublyCal is available at https://github.com/luyy9apples/DoublyCal.

3 Preliminaries

3.1 Uncertainty and Confidence

We extend the conceptualization of uncertainty and confidence for LLM outputs [Lin *et al.*, 2024] to general predictive systems. Given an input $\boldsymbol{x}$, a predictive system f produces a probability distribution over possible outputs $P_f(\boldsymbol{o} \mid \boldsymbol{x})$. The uncertainty of f regarding $\boldsymbol{x}$ is quantified by the dispersion (e.g., entropy) of this distribution. Conversely, the overall confidence of f can be defined inversely to this uncertainty. For a specific output $\boldsymbol{o}_i$, its confidence is directly associated with its assigned probability $P_f(\boldsymbol{o} = \boldsymbol{o}_i \mid \boldsymbol{x})$.

3.2 Knowledge Graph

A Knowledge Graph (KG) is a graph-structured database representing factual knowledge as a set of triples [Bollacker *et al.*, 2008]. Formally, a KG is denoted as $\mathcal{G} := (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{V}$ is a set of entities, $\mathcal{R}$ is a set of relations, and $\mathcal{E} := (h, r, t) \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is a set of factual triples. Each triple (h, r, t) represents an atomic fact, stating that relation r holds between head entity h and tail entity t. To mitigate the knowledge gaps of LLMs, KG-RAG provides them with structured evidence extracted from KGs [Xiang *et al.*, 2025].

3.3 Knowledge Graph Question Answering

Knowledge Graph Question Answering (KGQA) is a canonical knowledge-intensive reasoning task [Yih *et al.*, 2016]. Given a natural language question $\boldsymbol{Q}$ involving query entities $\mathcal{V}_{\boldsymbol{Q}}$, a reasoning system is expected to retrieve relevant evidence from a KG $\mathcal{G}$ and reason over it to produce the correct answer set $\mathcal{A}$. A standard knowledge-augmented pipeline (e.g., KG-RAG) involves two stages: (i) a retriever g that fetches a set of relevant evidence $\mathcal{Z}_{\boldsymbol{Q}} = g(\boldsymbol{Q}; \mathcal{G})$, and (ii) a reasoner f (e.g., an LLM) that predicts answers $\hat{\mathcal{A}} = f(\boldsymbol{Q}; \mathcal{Z}_{\boldsymbol{Q}}, \mathcal{G})$. This decomposition naturally highlights two distinct sources of uncertainty that our framework aims to calibrate: the evidence uncertainty in $\mathcal{Z}_{\boldsymbol{Q}}$, and the reasoning uncertainty in generating $\hat{\mathcal{A}}$ given $\mathcal{Z}_{\boldsymbol{Q}}$.

4 Methodology

This section introduces **DoublyCal**, a framework designed to establish a calibrated reasoning chain by jointly calibrating both verifiable KG evidence and the final LLM predictions. As illustrated in Figure 2, the proposed DoublyCal framework operates through three core components. Firstly, we formalize KG evidence as constrained relational paths and employ a Bayesian model to estimate a statistically grounded confidence for each evidence (Sec. 4.1). Then, a lightweight proxy model is trained under the supervision of these Bayesian confidence scores to generate KG evidence alongside its calibrated confidence (Sec. 4.2). Finally, the calibrated evidence-confidence pair serves as an objective signal integrated into any black-box LLM to mitigate its inherent overconfidence, thereby enhancing both the calibration and traceability of its final predictions (Sec. 4.3).

4.1 Bayesian Calibration of KG Evidence

KG Evidence Formulation. Effective KG evidence must balance *informativeness* for accurate reasoning with *interpretability* for reliable confidence estimation. While relational paths [Luo *et al.*, 2024] offer step-by-step interpretability, they may lack sufficient context. In contrast, subgraphs [Li *et al.*, 2025a] provide broader context but often introduce redundancy. To resolve this trade-off, we introduce *constrained relational paths* as our primary evidence form. This formulation augments a core relational path with an optional constraint derived from the neighborhood of the candidate answer, thereby enhancing informativeness while preserving interpretability.

Formally, given a KG $\mathcal{G}$ and a question $\boldsymbol{Q}$, a constrained relational path $\mathcal{P}_c$ is defined as the conjunction of a relational path $\mathcal{P}_r$ and an optional constraint $\mathcal{C}$:

$$\mathcal{P}_c := \mathcal{P}_r \, [\wedge \mathcal{C}] \,, \tag{1}$$

where $\mathcal{P}_r := \exists v_1, \ldots, v_{l-1}. r_1(q, v_1) \wedge r_2(v_1, v_2) \wedge \cdots \wedge r_l(v_{l-1}, \hat{a})$ denotes a directed relational path of length l from the query entity $q \in \mathcal{V}_{\boldsymbol{Q}}$ to a candidate answer $\hat{a}$. Here, each $r_i \in \mathcal{R}$ is a relation, and v_i is an existential variable. The optional constraint $\mathcal{C} := r_c(\hat{a}, c)$ represents a one-hop triple from the candidate answer $\hat{a}$ to a constraint entity $c \in \mathcal{V}$, which serves to filter or refine the candidate set.

Example. *Consider the question "What is the name of Snoopy's brother?" with the query entity $q =$ `Snoopy` and the true answer $a =$ `Spike`. Figure 2 illustrates a relational path evidence z_3 and its constrained counterpart z_2:*

$$z_3 := \mathrm{SiblingOf}(q, \hat{a}) \models_{\mathcal{G}} \{\texttt{Spike}, \texttt{Belle}\}, \tag{2}$$

$$z_2 := z_3 \wedge \mathrm{Gender}(\hat{a}, \texttt{Male}) \models_{\mathcal{G}} \{\texttt{Spike}\}, \tag{3}$$

where $\models_{\mathcal{G}}$ denotes grounding the evidence in $\mathcal{G}$ to obtain candidate answer entities. The auxiliary constraint on the candidate's gender effectively identifies $\hat{a} =$ `Spike`, yielding a more precise and informative evidence for reasoning.

Confidence Estimation with Beta-Bernoulli Model. To estimate a statistically grounded confidence for a given KG evidence $z_{\boldsymbol{Q}}$, we model it as a predictive system in accordance with the definition in Sec. 3.1. Formally, the system takes as input a candidate answer $\hat{a}$ drawn from the candidate set $[\![z_{\boldsymbol{Q}}]\!]$ obtained by grounding $z_{\boldsymbol{Q}}$ in $\mathcal{G}$ (i.e., $z_{\boldsymbol{Q}} \models_{\mathcal{G}} [\![z_{\boldsymbol{Q}}]\!]$), and produces a binary output indicating whether $\hat{a}$ is correct ($\hat{a} \in \mathcal{A}$). This defines a Bernoulli distribution for the correctness of a uniformly sampled candidate, characterized by the parameter $p \in [0, 1]$, which is precisely the probability that the candidate is correct.

To obtain a robust estimate of p that accounts for KG incompleteness, we impose a conjugate Beta prior $p \sim \mathrm{Beta}(\alpha, \beta)$, with hyperparameters $\alpha, \beta > 0$. Given $\boldsymbol{Q}$ and its answer set $\mathcal{A}$, we use the posterior mean as the calibrated confidence score p^*, which has the following closed form:

$$p^* = p(\mathcal{A} \mid z_{\boldsymbol{Q}}) = \frac{\alpha + \left|[\![z_{\boldsymbol{Q}}]\!] \cap \mathcal{A}\right|}{\alpha + \beta + \left|[\![z_{\boldsymbol{Q}}]\!]\right|}, \tag{4}$$

where $\left|[\![z_{\boldsymbol{Q}}]\!] \cap \mathcal{A}\right|$ counts the number of correct candidates, and $\left|[\![z_{\boldsymbol{Q}}]\!]\right|$ is the total number of grounded candidates. p^*

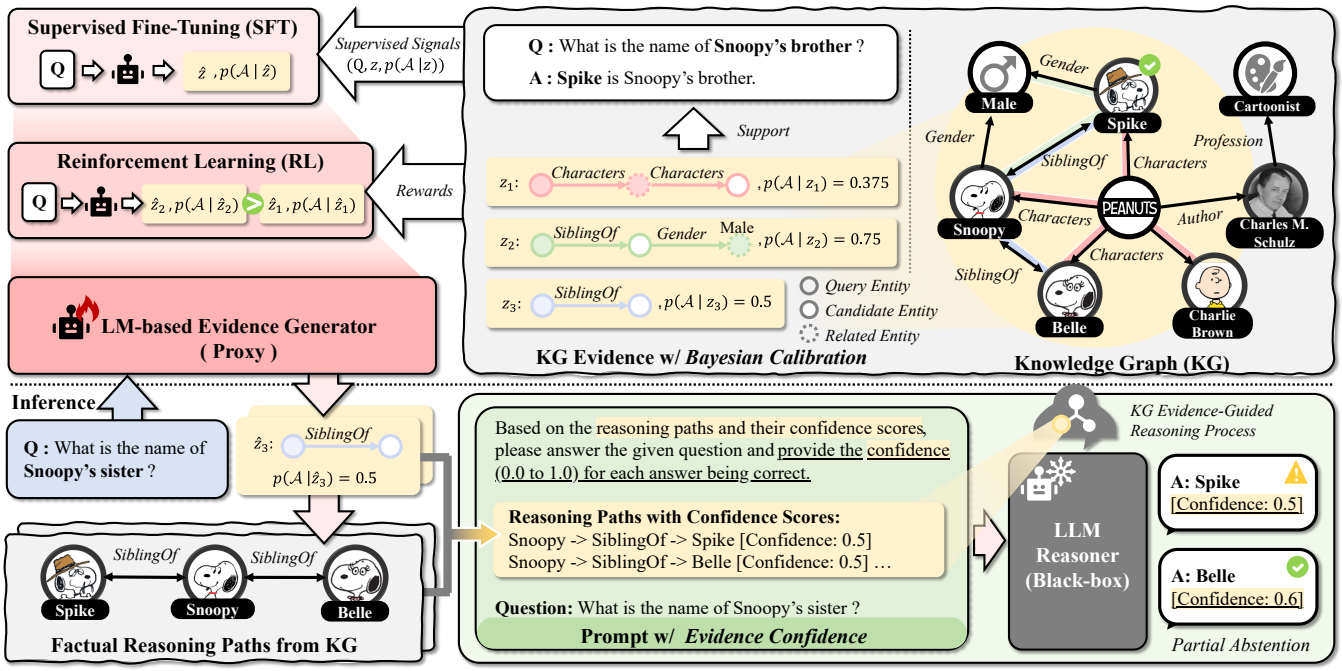

Figure 2: Overview of the DoublyCal framework. (**Top-right**) Bayesian calibration estimates statistically grounded confidence for KG evidence. (**Top-left**) A lightweight proxy model is trained to generate calibrated evidence-confidence pairs. (**Bottom**) During inference, these calibrated pairs guide a black-box LLM reasoner to produce final answers with well-calibrated confidence.

blends the empirical accuracy of the evidence with prior belief, mitigating the impact of sparse KG grounding (See Appendix A for detailed analysis).

Example (cont.). *With a weakly informative prior set to* $\alpha = \beta = 0.5$ *[Jeffreys, 1998], the statistical confidence for KG evidence* z_3 *and* z_2 *is estimated as* $p(\mathcal{A} \mid z_3) = 0.5$ *and* $p(\mathcal{A} \mid z_2) = 0.75$.

4.2 Proxy for Evidence Generation & Calibration

The Beta-Bernoulli model-based confidence provides a statistically grounded but *retrospective* measure of evidence quality. To enable *prospective* evidence retrieval and confidence estimation during inference, we introduce a lightweight **reasoning proxy**. This proxy model is designed to generate high-quality KG evidence alongside well-calibrated confidence estimates for any input question, thereby approximating a reliable reasoning path before the LLM's final prediction. The proxy model is implemented by an **LM-based Evidence Generator** and trained using the Bayesian confidence scores as its supervisory signal.

Supervised Fine-Tuning (SFT) for Evidence and Confidence Generation. We formalize the dual task of evidence generation and confidence estimation as a sequence-to-sequence problem and conduct the initial training of the proxy model via SFT. This stage equips the proxy with the fundamental ability to identify relevant KG evidence and estimate confidence by mimicking the Bayesian signal.

The SFT training dataset is constructed from triples $(\boldsymbol{Q}, z_{\boldsymbol{Q}}, p(\mathcal{A} \mid z_{\boldsymbol{Q}}))$. Each triple is formatted into a structured sequence using a predefined template: the question serves as the instruction, and the target output is the evidence path enclosed in XML-style tags, with the Bayesian confidence score included as an attribute (e.g., `<PATH confidence=...>...</PATH>`; see Appendix B.1 for details). The proxy model f_θ is trained via standard autoregressive language modeling to generate this target sequence (see Appendix C.1 for the objective).

Reinforcement Learning (RL) for Evidence Decision. We further refine the proxy by framing evidence generation and calibration as a sequential decision process optimized via RL. This stage transitions the proxy from imitation to strategic decision-making that jointly maximizes both inferential quality and confidence calibration.

For each question $\boldsymbol{Q}$ with a gold evidence set $\mathcal{Z}_{\boldsymbol{Q}}$, we compute a reward for the generated evidence $\hat{z}_{\boldsymbol{Q}}$ and its predicted confidence $\hat{c}$. Firstly, we define a match score $m(\hat{z}_{\boldsymbol{Q}}, z_{\boldsymbol{Q}}) \in [0, 1]$ for each $z_{\boldsymbol{Q}} \in \mathcal{Z}_{\boldsymbol{Q}}$, which combines Jaccard similarity [Jaccard, 1901] with an order-sensitive Levenshtein ratio [Levenshtein and others, 1966]. The overall reward R is a weighted combination of an inferential quality reward R_{inf} and a calibration alignment reward R_{cal}:

$$R = \lambda \cdot R_{\text{inf}} + (1 - \lambda) \cdot R_{\text{cal}}, \tag{5}$$

$$R_{\text{inf}} = \text{F1}(z_{\boldsymbol{Q}}) \cdot m(\hat{z}_{\boldsymbol{Q}}, z_{\boldsymbol{Q}}), \tag{6}$$

$$R_{\text{cal}} = \max\left(0, 1 - \xi \cdot |\hat{c} - c|\right), \tag{7}$$

$$\text{with } c = p(\mathcal{A} \mid z_{\boldsymbol{Q}}) \cdot m(\hat{z}_{\boldsymbol{Q}}, z_{\boldsymbol{Q}}). \tag{8}$$

Here, $\text{F1}(z_{\boldsymbol{Q}})$ is a precomputed F1 score assessing the reasoning capability of the gold evidence. Intuitively, a lower match score $m(\hat{z}_{\boldsymbol{Q}}, z_{\boldsymbol{Q}})$ reduces both the inferential quality

Reasoning Method	*KG Evidence*	+ UQ Method	WebQSP					CWQ				
			Hits	**Recall**	**F1**	**ECE ↓**	**ACE ↓**	**Hits**	**Recall**	**F1**	**ECE ↓**	**ACE ↓**
LLM Reasoner (GPT-3.5-turbo)	*No Augmentation*	+ Vanilla	74.7	53.1	44.6	27.7	26.6	47.7	40.3	29.5	38.8	38.0
		+ CoT	75.4	53.9	44.4	26.6	25.8	48.2	41.2	29.3	38.4	37.6
		+ Self-Probing	74.1	52.8	50.2	36.5	36.3	43.6	36.6	34.3	48.5	47.9
RoG [Luo *et al.*, 2024]	*Relational Path*	+ Vanilla	89.3	77.6	67.1	19.6	20.8	65.3	60.5	43.0	27.4	26.8
		+ CoT	89.9	78.5	68.2	18.9	17.9	65.3	60.4	43.7	27.6	27.0
		+ Self-Probing	87.5	76.6	73.5	13.9	12.8	61.9	56.9	48.7	38.0	37.4
SubgraphRAG [Li *et al.*, 2025a]	*Subgraph*	+ Vanilla	88.8	81.3	77.3‡	11.1	9.7	61.5	57.4	52.2‡	39.9	39.7
		+ CoT	89.0	81.0	77.1	10.6	9.5	59.4	55.7	51.4	38.9	38.7
		+ Self-Probing	89.6	80.7	74.9	12.3	12.8	59.9	56.0	50.2	39.1	38.0
SFT-DoublyCal (Ours)	*Constrained Relational Path*	+ Vanilla	90.0	81.0	72.6	3.1†	3.8‡	68.8	64.3	48.1	17.9	17.5
		+ CoT	90.1‡	81.3	72.1	3.5‡	3.6†	69.0	64.7	47.7	17.8‡	17.4
		+ Self-Probing	88.5	79.4	76.6	7.9	7.1	63.2	58.5	50.9	22.5	22.1
RL-DoublyCal (Ours)	*Constrained Relational Path*	+ Vanilla	91.5†	84.8‡	76.7	4.5	5.1	70.5‡	66.6‡	50.1	17.9	17.3‡
		+ CoT	91.5†	85.0†	76.8	3.9	4.3	71.3†	67.5†	49.8	17.6†	17.2†
		+ Self-Probing	89.9	83.0	79.3†	6.8	6.8	64.6	60.8	53.0†	23.5	23.1

Table 1: Main results (%) of our DoublyCal and the SingleCal baselines on WebQSP and CWQ datasets. Best, second-best, and worst results are highlighted in red †, green ‡, and gray, respectively.

of $\hat{z}_{\boldsymbol{Q}}$ and its target confidence c. The weight $\lambda \in (0, 1)$ balances the two objectives, and $\xi > 0$ is a tolerance coefficient. The final reward per generation is the maximum R over $\mathcal{Z}_{\boldsymbol{Q}}$, followed by transformations to ensure a smooth training signal. The policy π_θ of the proxy model is optimized to maximize the expected reward under the Group Relative Policy Optimization (GRPO) [Shao *et al.*, 2024] objective (see Appendices B.2 and C.2 for implementation details).

4.3 LLM Reasoning with Calibrated Evidence

The trained proxy equips any black-box primary LLM with double-calibration capability in a plug-and-play manner.

For a given question $\boldsymbol{Q}$, the proxy model generates a set of candidate evidence-confidence pairs, i.e., $\hat{\mathcal{Z}}_{\boldsymbol{Q}} = \{(\hat{z}^{(i)}_{\boldsymbol{Q}}, \hat{c}^{(i)})\}_{i=1\ldots K}$. Each $\hat{z}^{(i)}_{\boldsymbol{Q}}$ is grounded in $\mathcal{G}$, yielding factual reasoning paths that share the same confidence score $\hat{c}^{(i)}$. These paths and their confidences are verbalized into a natural-language context and integrated into prompts following prior verbalized UQ methods [Tian *et al.*, 2023; Xiong *et al.*, 2024]. Processing this enriched context, the LLM produces a final answer along with a well-calibrated prediction confidence. This design establishes a traceable chain of confidence and achieves double calibration: the proxy first calibrates the external evidence confidence, which then informs and refines the LLM's final prediction calibration through its own verbalized uncertainty estimation.

5 Experiments

5.1 Experimental Settings

Datasets and Evaluation Metrics. Following [Luo *et al.*, 2024; Li *et al.*, 2025a], we evaluate our framework on two widely-adopted Knowledge Graph Question Answering (KGQA) benchmarks: **WebQSP** [Yih *et al.*, 2016] and **CWQ** [Talmor and Berant, 2018]. To measure prediction accuracy, we report **Hits**, **Recall**, and macro-averaged **F1** score. To assess the reliability of confidence estimates, we report the **Expected Calibration Error (ECE)** [Guo *et al.*, 2017] and the **Adaptive Calibration Error (ACE)** [Nixon *et al.*, 2019]. ECE and ACE measure the expected absolute difference between empirical accuracy and predicted confidence using equal-width and adaptive equal-size bins, respectively.

Baselines. To rigorously evaluate our Double-Calibration mechanism, we construct **Single-Calibration (SingleCal)** baselines by extending reasoning paradigms with verbalized Uncertainty Quantification (UQ) methods, which elicit a self-reported confidence score alongside the predicted answer. We select three state-of-the-art reasoning frameworks: (i) The base **LLM Reasoner** without KG access; (ii) **RoG** [Luo *et al.*, 2024], a KG-RAG method that grounds reasoning in retrieved relational paths; (iii) **SubgraphRAG** [Li *et al.*, 2025a], which retrieves and reasons over KG subgraphs. Each framework is combined with three representative UQ prompting techniques: **Vanilla** [Tian *et al.*, 2023], **CoT** [Kojima *et al.*, 2022], and **Self-Probing** [Xiong *et al.*, 2024]. Prompt templates are detailed in Appendix D.

Implementation Details. All evaluated methods employ GPT-3.5-turbo [Floridi and Chiriatti, 2020] as the primary reasoner unless otherwise specified, ensuring that performance differences are directly attributable to the calibration mechanism rather than the base LLM capability. The evidence proxy in our DoublyCal is implemented with Llama2-7B-Chat [Touvron *et al.*, 2023], which is trained via the SFT then RL pipeline described in Sec. 4.2. More details of experimental settings are provided in Appendix E.

5.2 Main Results

Table 1 summarizes the comparative performance of our DoublyCal against all SingleCal baselines.

Superiority of Double-Calibration. DoublyCal achieves the best overall performance, consistently leading in all prediction metrics (Hits, Recall, F1) and the lowest ECE and

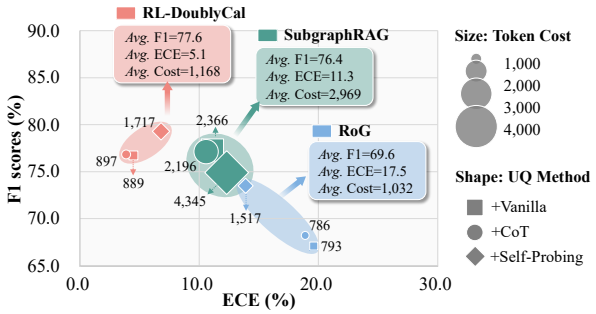

Figure 3: Input token efficiency analysis on WebQSP.

Variant	Hits	Recall	F1	ECE ↓	ACE ↓
SFT Only					
DoublyCal	90.0	81.0	72.6	3.1	3.8
SingleCal	90.1 (+0.1)	80.4 (-0.6)	72.5 (-0.1)	21.2 (+18)	20.3 (+16)
Evidence	83.7 (-6.3)	80.2 (-0.8)	62.5 (-10)	21.2 (+18)	21.0 (+17)
With RL					
DoublyCal	91.5	84.8	76.7	4.5	5.1
SingleCal	91.6 (+0.1)	84.2 (-0.6)	75.8 (-0.9)	20.6 (+16)	20.0 (+15)
Evidence	86.4 (-5.1)	84.8	67.8 (-8.9)	21.3 (+16)	21.1 (+16)

Table 2: Ablation study results (%) on the WebQSP dataset with the Vanilla UQ method. Gray marks performance degradation relative to the full model.

ACE. Notably, it establishes a new standard for reliability, reducing the ECE and ACE to levels significantly lower than all SingleCal baselines. This result demonstrates that while KG-RAG methods can enhance LLM factuality, calibrating *both* the external KG evidence and the final prediction is necessary for achieving reliable reasoning. Furthermore, the observed gains in prediction metrics indicate that evidence confidence helps improve the quality of the retrieved evidence and further unlocks the LLM's reasoning potential.

Calibrated Evidence as an Anchor for Verbalized UQ. The efficacy of verbalized UQ methods varies across reasoning backbones, with none proving universally dominant. This inconsistency arises because these methods may be subject to LLMs' inherent overconfidence. DoublyCal addresses this by supplying KG evidence with calibrated confidence, providing a reliable external anchor that refines the LLM's uncertainty expression. Consequently, DoublyCal stabilizes all three UQ techniques and achieves the lowest ECE and ACE in nearly every configuration, showing that externally calibrated evidence improves confidence elicitation.

Controlled Enhancement via RL. The RL stage yields a significant performance gain, improving average F1 by $\sim$3.0 percentage points over the SFT-only. While prior work notes RL's risk of harming calibration [Kalai *et al.*, 2025], our Bayesian confidence-aligned reward successfully mitigates this trade-off, resulting in only a minor and controlled variation in ECE and ACE. This confirms that our reward design effectively balances accuracy and calibration.

5.3 Efficiency Analysis

We further analyze the input token efficiency of each method, with results shown in Figure 3.

Superior Cost-Effectiveness of DoublyCal. DoublyCal substantially outperforms RoG (F1 +8.0, ECE -12.4) with only a marginal increase in input token cost, while consuming only about 39% of the input tokens required by SubgraphRAG. This efficiency stems from the high information density of the evidence provided by DoublyCal. Specifically, compared to the simple relational paths in RoG, our constrained relational paths incorporate an optional constraint that yields more precise and informative evidence without compromising conciseness. Moreover, our proxy model is trained through evidence confidence calibration to select more discriminative KG evidence, further enhancing retrieval precision. In contrast, while SubgraphRAG's subgraphs offer broader context, their lower information density leads to disproportionately high token costs.

Efficiency Across Different UQ Methods. Self-Probing incurs approximately twice the input token cost of Vanilla or CoT due to its two-step prompting design. However, because DoublyCal and RoG retrieve concise evidence, they can effectively leverage Self-Probing's reflective "second-thought" process without excessive overhead. Notably, even when equipped with Self-Probing, their total input token cost remains below that of SubgraphRAG+Vanilla.

5.4 Ablation Analysis

We ablate DoublyCal against two variants (Table 2) to examine the contribution of each component: (i) **SingleCal**, which removes the calibrated evidence confidence and applies calibration only to the final LLM output; (ii) **Evidence**, which removes the LLM reasoner and directly outputs the terminal entity of the factual path ($[\![z_Q]\!]$) as the answer, using the evidence confidence as the final confidence score.

Evidence Confidence is Crucial for Final Prediction Calibration. Ablating from DoublyCal to SingleCal reveals a stark outcome: while predictive accuracy remains stable (e.g., F1 changes within $\pm$1 point), the calibration error (ECE and ACE) increases drastically from $\sim$4 to $>$20. This indicates that externally calibrated evidence confidence is essential, as it provides the LLM with a reliable anchor for its self-assessment. Without this first-stage calibration, the black-box LLM cannot reliably judge its own certainty, even when it can identify correct answers using high-quality KG evidence.

The LLM Reasoner Enables Integrative Reasoning. The significant performance gap of the Evidence variant underscores a pivotal design insight: the evidence proxy and the LLM reasoner play distinct yet complementary roles. The proxy specializes in evaluating individual KG evidence, while the LLM reasoner excels at synthesizing an ensemble of such evidence to perform complex reasoning. Consequently, the final prediction confidence is not a simple pass-through of any single evidence confidence, but rather the result of the LLM's holistic reasoning over the entire set of calibrated evidence. This efficient proxy-reasoner synergy is essential to the framework's performance.

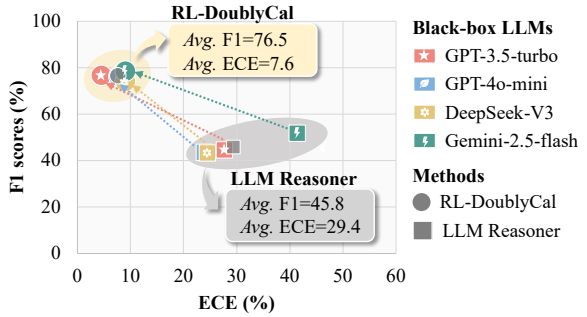

Figure 4: Prediction accuracy (F1) and calibration (ECE) of diverse black-box LLMs with and without DoublyCal.

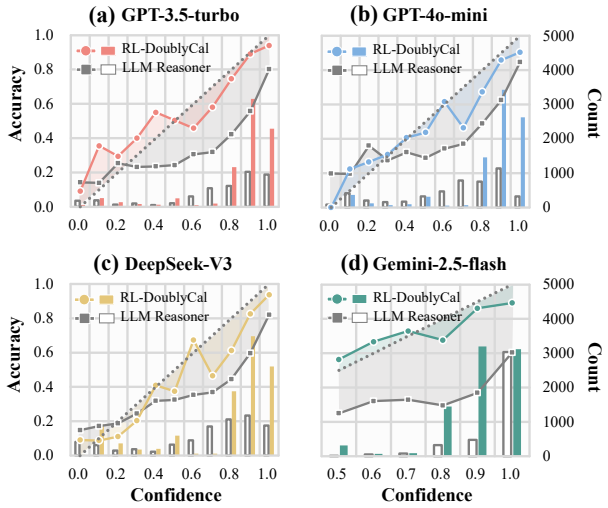

Figure 5: Calibration diagrams (*bars*: confidence distribution per confidence bin; *line*: empirical accuracy; *dashed*: ideal calibration).

5.5 Cross-model Compatibility Analysis

To assess generalizability of DoublyCal, we evaluate it across diverse black-box LLMs, including GPT-3.5-turbo, GPT-4o-mini [Achiam *et al.*, 2023], DeepSeek-V3 [Liu *et al.*, 2024], and Gemini-2.5-flash [Comanici *et al.*, 2025].

Performance-Reliability Trade-off in LLM Reasoners. Figure 4 reveals a clear trade-off between accuracy and calibration among standalone LLMs. While GPT-family models and DeepSeek achieve comparable accuracy with moderate calibration errors (F1: 43.3–44.6; ECE: 23.9–27.7), Gemini attains a notably higher F1 (51.8) at the cost of a significantly worse ECE (41.4). This pattern highlights a common pitfall where optimizing purely for accuracy often degrades reliability in standalone LLMs. Confidence distributions (Figure 5) confirms that all models exhibit systematic overconfidence. This issue is most acute in Gemini, where roughly 80% of predictions are made with maximal confidence (1.0), yet the accuracy within this high-confidence group is only about 0.6.

DoublyCal Systematically Decouples the Trade-off. DoublyCal delivers consistent and substantial improvements across all models, effectively decoupling this trade-off. As shown in Figure 4, it raises the average F1 from 45.8 to 76.5 while reducing the average ECE from 29.4 to 7.6. Crucially, DoublyCal mitigates the overconfidence patterns observed in black-box LLMs (Figure 5), shifting confidence distributions toward well-calibrated and high-accuracy regions. By grounding confidence in externally calibrated evidence, DoublyCal provides a generalizable solution that enhances both accuracy and reliability of diverse black-box LLMs.

Sample
Question: Where did George W. Bush live as a child?
Query Entity (q)**:** George W. Bush **Answers:** New Haven.
RL-DoublyCal + Self-Probing
Retrieval:
• q → people.person.place_of_birth → New Haven [Confidence: 0.8]
• q → people.person.place_of_birth → New Haven → location.location.containedby → *Connecticut* [Confidence: 0.5]
• q → people.person.place_of_birth → New Haven → location.location.containedby → United States of America [Confidence: 0.5]
Predictions: {*Connecticut*: 0.3}
SubgraphRAG + Vanilla
Retrieval:
• (q, people.person.place_of_birth, New Haven)
• (q, people.person.nationality, United States of America)
• (m.03prwzr, people.place_lived.location, *Midland*)
• (m.02xlp0j, people.place_lived.location, Washington, D.C.) • ...
Predictions: {*Midland*: 1.0}

Table 3: Case study: DoublyCal vs. SingleCal baseline.

Sample
Question: Where was Martin Luther King, Jr. raised?
Query Entity (q)**:** Martin Luther King, Jr. **Answers:** Atlanta.
RL-DoublyCal (Full) + Vanilla
Retrieval:
• q people.person.place_of_birth → *Atlanta* [Confidence: 0.8]
• q → people.deceased_person.place_of_death → *Memphis* [Confidence: 0.8]
Predictions: {*Atlanta*: 0.8, *Memphis*: 0.1}
RL-DoublyCal (SingleCal) + Vanilla
Retrieval: Same factual paths as above, without confidence scores
Predictions: {*Atlanta*: 0.8, *Memphis*: 0.2}

Table 4: Case study: DoublyCal vs. its SingleCal variant.

5.6 Case Studies

This section presents two case studies that qualitatively illustrate how DoublyCal enhances reliability of LLM predictions.

As shown in Table 3, for a question lacking a direct KG relation ("lived as a child") to provide accurate support, both methods retrieve the related birthplace fact. However, while SubgraphRAG retrieves scattered evidence leading to an overconfident error (confidence 1.0), DoublyCal presents concise paths with calibrated confidence scores. Its birthplace path receives high confidence (0.8), while less relevant expansions are scored lower (0.5). Consequently, the LLM correctly assigns low confidence (0.3) to the incorrect answer "Connecticut". Furthermore, the ablation study in Table 4 demonstrates that explicitly providing evidence confidence makes the LLM's predicted confidence more concentrated on the correct answer. Together, these cases show DoublyCal's ability to mitigate overconfidence and improve calibration. Appendix F provides complementary experiments.

6 Conclusion

This paper establishes the principle of double-calibration for constructing a calibrated reasoning chain from KG evidence retrieval to final LLM prediction. We implement this principle in DoublyCal, a reliable KG-RAG framework that integrates plug-and-play verbalized uncertainty quantification, thereby enhancing the traceability and reliability of diverse black-box LLMs. Our work offers a concrete step toward building more reliable and transparent LLM systems, contributing to the advancement of trustworthy AI.

Limitations

The proposed Double-Calibration principle is task-agnostic and readily adaptable to KG-RAG, and its effectiveness is well supported by our experiments. However, the robustness of DoublyCal still leaves room for improvement. On the one hand, while the Beta-Bernoulli confidence estimator can partially alleviate the effect of incomplete KGs via Bayesian smoothing, the resolution of its estimated confidence may decline when the KG is extremely sparse. On the other hand, although constrained relational paths provide strong generality, the fixed form of KG evidence somewhat limits DoublyCal's performance on complex questions, such as those involving aggregation or comparison. Future work with richer evidence forms or paradigms that iteratively explore the KG and dynamically construct evidence (e.g., via agentic planning) could further increase flexibility and reduce the reasoning burden placed on the black-box LLM.

References

[Achiam *et al.*, 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[Bayes, 1958] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Biometrika*, 45(3-4):296–315, 1958.

[Bollacker *et al.*, 2008] Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[Chen *et al.*, 2024] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: llms' internal states retain the power of hallucination detection. In *ICLR*, 2024.

[Comanici *et al.*, 2025] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261, 2025.

[Floridi and Chiriatti, 2020] Luciano Floridi and Massimo Chiriatti. GPT-3: Its nature, scope, limits, and consequences. *Minds Mach.*, 30(4):681–694, 2020.

[Guo *et al.*, 2017] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, pages 1321–1330, 2017.

[Guo *et al.*, 2025] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In *Findings of EMNLP*, pages 10746–10761, 2025.

[Haldane, 1932] John Burdon Sanderson Haldane. A note on inverse probability. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 28, pages 55–61. Cambridge University Press, 1932.

[He *et al.*, 2024] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *NeurIPS*, 2024.

[Huang *et al.*, 2024] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John C. Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. Position: Trustllm: Trustworthiness in large language models. In *ICML*, 2024.

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

[Jaccard, 1901] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat.*, 37:547–579, 1901.

[Jeffreys, 1998] Harold Jeffreys. *The theory of probability*. OuP Oxford, 1998.

[Jiang *et al.*, 2023] Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *ICLR*, 2023.

[Kalai *et al.*, 2025] Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate. *CoRR*, abs/2509.04664, 2025.

[Kojima *et al.*, 2022] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

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

[Levenshtein and others, 1966] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710, 1966.

[Li *et al.*, 2025a] Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *ICLR*, 2025.

[Li *et al.*, 2025b] Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. In *ICLR*, 2025.

[Lin *et al.*, 2024] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Trans. Mach. Learn. Res.*, 2024, 2024.

[Liu *et al.*, 2024] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024.

[Liu *et al.*, 2026] Shuyi Liu, Yuming Shang, and Xi Zhang. Truthfulrag: Resolving factual-level conflicts in retrieval-augmented generation with knowledge graphs. In *AAAI*, pages 32168–32176, 2026.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[Luo *et al.*, 2024] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *ICLR*, 2024.

[Malinin and Gales, 2021] Andrey Malinin and Mark J. F. Gales. Uncertainty estimation in autoregressive structured prediction. In *ICLR*, 2021.

[Manakul *et al.*, 2023] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *EMNLP*, pages 9004–9017, 2023.

[Mavromatis and Karypis, 2025] Costas Mavromatis and George Karypis. GNN-RAG: graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of ACL*, pages 16682–16699, 2025.

[Nixon *et al.*, 2019] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *CVPR Workshops*, number 7, 2019.

[Pan *et al.*, 2024] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Trans. Knowl. Data Eng.*, 36(7):3580–3599, 2024.

[Shao *et al.*, 2024] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

[Stangel *et al.*, 2025] Paul Stangel, David Bani-Harouni, Chantal Pellegrini, Ege Özsoy, Kamilia Zaripova, Matthias Keicher, and Nassir Navab. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. *CoRR*, abs/2503.02623, 2025.

[Sun *et al.*, 2025] Jiashuo Sun, Xianrui Zhong, Sizhe Zhou, and Jiawei Han. Dynamicrag: Leveraging outputs of large language model as feedback for dynamic reranking in retrieval-augmented generation. In *NeurIPS*, 2025.

[Talmor and Berant, 2018] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *NAACL-HLT*, pages 641–651, 2018.

[Tanneru *et al.*, 2024] Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. Quantifying uncertainty in natural language explanations of large language models. In *AISTATS*, pages 1072–1080, 2024.

[Team, 2025] Qwen Team. Qwen3 technical report, 2025.

[Tian *et al.*, 2023] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *EMNLP*, pages 5433–5442, 2023.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[Vazhentsev *et al.*, 2025] Artem Vazhentsev, Lyudmila Rvanova, Ivan Lazichny, Alexander Panchenko, Maxim

Panov, Timothy Baldwin, and Artem Shelmanov. Token-level density-based uncertainty quantification methods for eliciting truthfulness of large language models. In *NAACL*, pages 2246–2262, 2025.

[Xia *et al.*, 2025] Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. A survey of uncertainty estimation methods on large language models. In *Findings of ACL*, pages 21381–21396, 2025.

[Xiang *et al.*, 2025] Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in RAG: A comprehensive analysis for graph retrieval-augmented generation. *CoRR*, abs/2506.05690, 2025.

[Xiong *et al.*, 2024] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *ICLR*, 2024.

[Yih *et al.*, 2016] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *ACL*, 2016.

[Zhang *et al.*, 2018] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *AAAI*, 2018.

[Zhang *et al.*, 2022] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *ACL*, pages 5773–5784, 2022.

[Zhang *et al.*, 2025a] Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. A survey of graph retrieval-augmented generation for customized large language models. *CoRR*, abs/2501.13958, 2025.

[Zhang *et al.*, 2025b] Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. Faithfulrag: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In *ACL*, pages 21863–21882, 2025.

A The Necessity of Bayesian Smoothing

As described in Sec. 4.1, we employ a Bayesian Beta-Bernoulli model to estimate KG evidence confidence. This appendix provides the justification for this design, demonstrating why the adopted Bayesian approach is essential for achieving robust estimates.

The Small-Sample Challenge. The structural sparsity inherent in knowledge graphs ($|\mathcal{E}| \ll |\mathcal{V}|^2 \times |\mathcal{R}|$) means that a given evidence $z_{\boldsymbol{Q}}$ for a question $\boldsymbol{Q}$ often retrieves only a small set of candidate answers. Formally, the candidate set size $n = |[\![z_{\boldsymbol{Q}}]\!]|$ is low. For instance, evidence such as $z : \text{SiblingOf}(q, \hat{a})$ typically yields $n \leq 5$ candidates. Consequently, it is frequent to encounter extreme sampling outcomes where the number of correct candidates, $s = |[\![z_{\boldsymbol{Q}}]\!] \cap \mathcal{A}|$, is either 0 or n.

In such extreme situations with small samples, the standard maximum-likelihood estimator (MLE), $\hat{p}_{MLE} = s/n$, collapses to an absolute and potentially misleading value:

$$\hat{p}_{MLE} = \begin{cases} 0, & s = 0 \\ 1, & s = n. \end{cases}$$

This behavior is illustrated in Figure 6(a). Such estimates are statistically unstable and semantically unreliable for real-world KGs, often causing overconfidence in incorrect answers or undue dismissal of correct ones.

Bayesian Smoothing with the Jeffreys Prior. To mitigate this instability, we adopt a Bayesian approach. By introducing a conjugate Beta prior distribution for the parameter p, the point estimate is naturally shrunk toward the prior mean, thereby balancing the observed empirical frequency with a prior belief. A principled choice for this prior is the Jeffreys prior, $\text{Beta}(0.5, 0.5)$ [Jeffreys, 1998], which is theoretically well-motivated as a "non-informative" reference. Under this prior, the posterior mean estimator (Eq. (4) in the main text) takes the form:

$$p^* = \frac{0.5 + s}{1 + n}.$$

For the extreme cases, this becomes:

$$p^* = \begin{cases} \frac{0.5}{1+n}, & s = 0, \\ \frac{0.5+n}{1+n}, & s = n. \end{cases}$$

This formulation provides automatic and theoretically grounded smoothing. As shown in Figure 6(b), when n is small, the estimate is conservatively pulled toward 0.5, guarding against overconfidence. As n grows, p^* converges to the MLE ($\hat{p}_{MLE}$), ensuring the estimator's consistency.

It is worth noting that the choice of prior controls the strength of this smoothing. For instance, the uniform prior $\text{Beta}(1, 1)$ [Bayes, 1958] yields the estimator $p^* = (1 + s)/(2 + n)$ (Figure 6(c)), which exerts a stronger shrinkage effect toward 0.5 than the Jeffreys prior and thus tends to produce more conservative estimates. These theoretical considerations are validated and their practical impact compared through an empirical analysis provided in Appendix F.4.

B Prompt Templates for the Proxy Model

This appendix details the input-output formats used for training the proxy model.

B.1 Supervised Fine-Tuning (SFT)

In the SFT stage, the proxy model learns to predict a target output sequence autoregressively. Each training instance consists of a natural-language instruction containing the question, paired with a target sequence that encodes the corresponding KG evidence in an XML-style format.

To formally describe the template, we define the symbolic placeholders: `[CONFIDENCE_SCORE]` denotes the Bayesian confidence; `[RELATION_PATH]` represents the core relational path, with individual relations separated by the special token `<SEP>`. Within the optional `<CONSTRAINT>` block, `[CONSTRAINED_REL_ENT]` signifies the concatenation of a constraining relation and its corresponding entity, also joined by `<SEP>`. The SFT template and a concrete example are provided below.

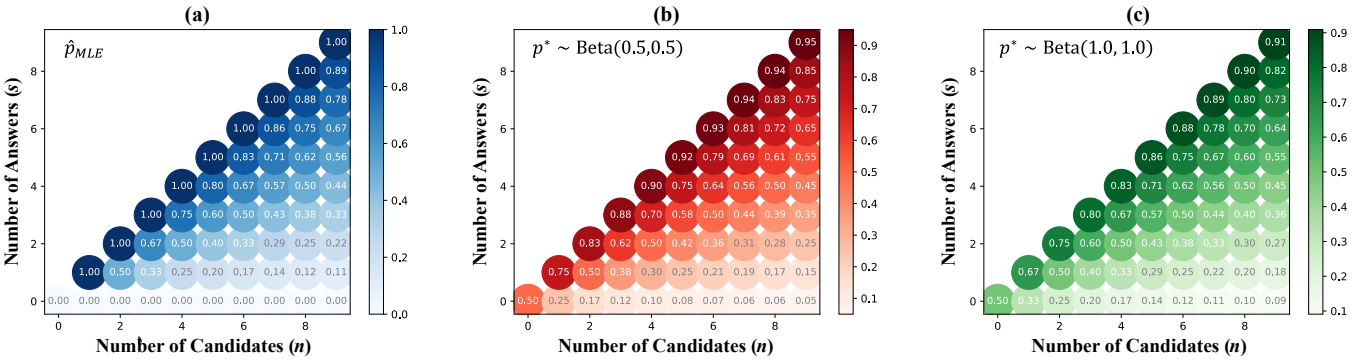

Figure 6: Comparison of **(a)** MLE, **(b)** Jeffreys-prior posterior mean, and **(c)** Uniform-prior posterior mean across varying candidate sizes n and correct counts s.

B.2 Reinforcement Learning (RL)

In the RL phase, the proxy model is prompted to generate an enhanced KG evidence path with well-calibrated confidence. The input instruction is adapted to encourage strategic decision-making, while the target output format remains identical to that used in the SFT stage.

SFT Template

Input: `Please generate a valid relation path that can be helpful for answering the following question: [QUESTION]`

Expected Output (with constraint):

```
<PATH confidence=[CONFIDENCE_SCORE]>
[RELATION_PATH]<CONSTRAINT>
[CONSTRAINED_REL_ENT]
</CONSTRAINT></PATH>
```

Expected Output (without constraint):

```
<PATH confidence=[CONFIDENCE_SCORE]>
[RELATION_PATH]</PATH>
```

SFT Example

Input: `Please generate a valid relation path that can be helpful for answering the following question: what is the name of snoopy's brother?`

Expected Output:

```
<PATH confidence=0.75>sibling_of
<CONSTRAINT>gender<SEP>male
</CONSTRAINT></PATH>
```

RL Template

Input: `Please generate an enhanced relation path with well-calibrated confidence that can be helpful for answering the following question: [QUESTION]`

RL Example

Input: `Please generate an enhanced relation path with well-calibrated confidence that can be helpful for answering the following question: what is the name of snoopy's brother?`

The proxy model must then generate a full output sequence (e.g., `<PATH confidence=...>...</PATH>`) based on this instruction. The generated sequence is subsequently evaluated by the reward function described in Sec. 4.2, which jointly assesses the inferential quality of the evidence path and the calibration accuracy of its attached confidence score.

C Training Details of the Proxy Model

This appendix details the training objectives and implementation for the SFT then RL training of the proxy model.

C.1 SFT Stage

In the SFT stage, the proxy model f_θ is trained to autoregressively generate the target structured sequence (i.e., KG evidence with Bayesian confidence). The objective is to minimize the standard cross-entropy loss over the token sequence:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{t=1}^{T} \log P_\theta(o_t \mid o_{<t}, \boldsymbol{Q}),$$

where $\boldsymbol{o} = (o_1, \ldots, o_T)$ is the token sequence of the target output, and $P_\theta(o_t \mid o_{<t}, \boldsymbol{Q})$ is the probability predicted by f_θ for the t-th token given the input question $\boldsymbol{Q}$ and previous tokens $o_{<t}$.

C.2 RL Stage

Final Reward Function. The reward defined in Eq. (5) is a weighted sum of the inferential quality reward R_{inf} and the calibration alignment reward R_{cal}, originally bounded in $[0, 1]$. To stabilize optimization, we map the raw reward into the continuous interval $[-1, 2]$ using a sigmoid-shaped transformation, which is defined as follows:

$$R' = 3 \cdot \sigma\big(\xi' \cdot (R - 0.5)\big) - 1,$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\xi' > 0$ is a scaling hyperparameter (set to $\xi' = 2$ in our experiments). Additionally, we introduce a penalty of -3 for syntactically invalid outputs, such as when the generated sequence does not contain the required `<PATH>` tag.

GRPO Policy Objective. The proxy's policy π_θ is optimized by minimizing the Group Relative Policy Optimization (GRPO) loss [Shao *et al.*, 2024]. This objective encourages higher reward while preventing excessive deviation from the reference policy (the model after SFT), thereby maintaining generation quality and training stability:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\mathbb{E}_{(s,\boldsymbol{o}_i)}$$
$$\left[\frac{\pi_\theta(\boldsymbol{o}_i \mid s)}{[\pi_\theta(\boldsymbol{o}_i \mid s)]_{\text{no grad}}}\hat{A}_i \;-\; \beta'\,\text{KL}\big[\pi_\theta \| \pi_{\text{ref}}\big]\right],$$

where s denotes the shared input prompt for a group of size G, $\boldsymbol{o}_i$ is the i-th generated output sequence in the group, π_{ref} is the reference policy (the model after SFT), $\hat{A}_i$ is the estimated advantage for the sequence $\boldsymbol{o}_i$, and β' controls the strength of the KL regularization term.

D Prompt Templates for UQ Methods

This appendix details the prompt templates used for the three verbalized Uncertainty Quantification (UQ) methods evaluated in our work: **Vanilla** [Tian *et al.*, 2023], **CoT** [Kojima *et al.*, 2022], and **Self-Probing** [Xiong *et al.*, 2024]. Each template is designed to elicit answers along with calibrated confidence estimates from a black-box LLM.

D.1 Vanilla Template

The Vanilla template directly instructs the model to output answers with confidence scores in a specified JSON format.

Vanilla Template

Input: `[KG_RAG_INSTRUCTION] Please answer the following questions and provide the confidence (0.0 to 1.0) for each answer being correct. Please keep the answer as simple as possible and return all the possible answers and their confidence as a json string. Output format example: {<answer_1>: <confidence_1>,...,<answer_k>: <confidence_k>} [KG_RAG_CONTEXT] Question: [QUESTION]`

D.2 Chain-of-Thought (CoT) Template

The CoT template extends the Vanilla approach by appending the instruction "`Let's think it step by step.`" before presenting the context and question, thereby encouraging the model to generate an explicit reasoning chain prior to providing the final answer and confidence.

CoT Template

Input: `[KG_RAG_INSTRUCTION] Please answer the following questions and provide the confidence (0.0 to 1.0) for each answer being correct. Please keep the answer as simple as possible and return all the possible answers and their confidence as a json string. Output format example: {<answer_1>: <confidence_1>,...,<answer_k>: <confidence_k>} Let's think it step by step. [KG_RAG_CONTEXT] Question: [QUESTION]`

D.3 Self-Probing Template

The Self-Probing method employs a two-round dialogue. The first prompt elicits a list of candidate answers. The LLM's generated answer list is then used in a second prompt, which instructs it to analyze the likelihood of each answer being correct and to output the corresponding confidence scores in the same JSON format.

Self-Probing Template

First Interaction (Answer Generation):

```
[KG_RAG_INSTRUCTION] Please answer
the given question. Please keep
the answer as simple as possible
and return all the possible answers
as a list.
[KG_RAG_CONTEXT]
Question: [QUESTION]
```

Model Output: `[ANSWER_LIST]`

Self-Probing Template (Cont.)

Second Interaction (Confidence Elicitation):

```
Q: How likely are the above answers to
be correct? Analyze the possible
answers, provide your reasoning
concisely, and give your confidence
(0.0 to 1.0) for each answer being
correct. Please keep the answer
as simple as possible and return
all the possible answers and their
confidence as a json string.
Output format example: {<answer_1>:
<confidence_1>,...,<answer_k>:
<confidence_k>}
```

E Details of Experimental Settings

E.1 Datasets

Our main experiments are conducted on two established KGQA benchmarks: **WebQSP** [Yih *et al.*, 2016] and **CWQ** [Talmor and Berant, 2018], both of which are based on the Freebase knowledge graph [Bollacker *et al.*, 2008]. To ensure fair comparison, we adopt the same train/validation/test splits used in prior work [Luo *et al.*, 2024; Li *et al.*, 2025a]. The detailed statistics of both datasets are presented in Table 5.

Dataset	#Train	#Validation	#Test	Max #Hop
WebQSP	2,826	225	1,628	2
CWQ	27,639	2,577	3,531	4

Table 5: Statistics of the Freebase datasets.

To further verify the robustness of DoublyCal across different KG schemas, we evaluate it on the **MetaQA** dataset [Zhang *et al.*, 2018], which is constructed from WikiMovies and employs a separate KG. The statistics of MetaQA are summarized in Table 6. Due to computational resource constraints, we randomly sample 1,000 questions from the original test set of each MetaQA split for evaluation.

Dataset	#Train	#Validation	#Test	Sampled #Test
MetaQA-1hop	96,106	9,992	9,947	1,000
MetaQA-2hop	118,980	14,872	14,872	1,000
MetaQA-3hop	114,196	14,274	14,274	1,000

Table 6: Statistics of the MetaQA datasets.

E.2 Evaluation Metrics

To evaluate KGQA performance, we follow [Luo *et al.*, 2024] and report **Hits**, **Recall**, and **F1**. For a question $\boldsymbol{Q}$ with gold answer set $\mathcal{A}$:

- **Hits** indicates whether the predicted answer set $\hat{\mathcal{A}}$ contains at least one correct answer:

$$\text{Hits} = \mathbf{1}\big[|\hat{\mathcal{A}} \cap \mathcal{A}| > 0\big], \tag{9}$$

  where $\mathbf{1}[\cdot]$ is the indicator function.

- **Recall** measures the fraction of gold answers covered by the prediction:

$$\text{Recall} = \frac{|\hat{\mathcal{A}} \cap \mathcal{A}|}{|\mathcal{A}|}. \tag{10}$$

- **F1** is the harmonic mean of Precision and Recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{11}$$

  where $\text{Precision} = |\hat{\mathcal{A}} \cap \mathcal{A}| \,/\, |\hat{\mathcal{A}}|$ is the fraction of predicted answers that are correct.

To evaluate confidence calibration, we use the Expected Calibration Error (**ECE**) and Adaptive Calibration Error (**ACE**). Following the standard definition of **ECE** in [Guo *et al.*, 2017], we partition all predictions into M equal-width bins according to their predicted confidence. In our experiments, we set $M = 10$. Formally,

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \big|\text{acc}(B_m) - \text{conf}(B_m)\big|, \tag{12}$$

where N is the total number of samples, B_m denotes the set of predictions falling into the m-th bin, $\text{acc}(B_m)$ denotes the accuracy within that bin, and $\text{conf}(B_m)$ denotes the average predicted confidence in the bin.

ACE follows the same formulation but employs an adaptive binning scheme [Nixon *et al.*, 2019]: predictions are sorted by confidence and partitioned into M equal-size bins (we also use $M = 10$).

E.3 Implementation Details

SFT Training Dataset Construction. We construct the SFT training data through the following pipeline. For each question $\boldsymbol{Q}$, a breadth-first search (max depth$= 4$) identifies the shortest relational paths $\mathcal{P}_r$ between the query entity q and each candidate answer a. The Bayesian confidence for each $\mathcal{P}_r$ is computed as per Sec. 4.1. To enhance evidence quality, we then gather potential constraints $\mathcal{C}$ from each answer's one-hop neighborhood. For each candidate constrained path $\mathcal{P}_c = \mathcal{P}_r \wedge \mathcal{C}$, we compute its Bayesian confidence and retain

it only if $\mathcal{P}_c$ yields a higher confidence than $\mathcal{P}_r$ alone. This filtering trains the proxy to identify genuinely valuable constraints. The SFT stage follows RoG's multi-task setup [Luo *et al.*, 2024], jointly optimizing the primary evidence generation and calibration task alongside an auxiliary QA task.

Training and Inference. The proxy model is trained using the AdamW optimizer [Loshchilov and Hutter, 2019]. We set the learning rate to 2×10^{-5} for the SFT stage and 1.41×10^{-5} for the RL stage. Each stage is trained for at most 3 epochs with early stopping. All experiments were conducted on two NVIDIA A100 (80GB) GPUs. Training DoublyCal requires approximately 8.5 hours for SFT and 3 hours for RL, which is a lightweight one-time cost that enables efficient inference with any black-box LLM.

At inference time, the proxy model uses greedy decoding (temperature $\tau = 0$) to deterministically generate the top-K evidence. Following RoG, we set $K = 3$, and the model produces constrained relational paths with a maximum depth of 4. For each question, this generation step takes around 1 second on an A100 GPU. The resulting paths are then grounded against the KG via efficient entity and relation lookups. This generative retrieval paradigm avoids expensive full-graph traversal and scales sublinearly with KG size.

Hyperparameter Selection. All key hyperparameters were set based on established design principles and empirical observations from pilot studies, given the computational cost of exhaustive grid search. For Bayesian calibration (Eq. (4)), we adopt the weakly informative Jeffreys prior with $\alpha = \beta = 0.5$ [Jeffreys, 1998]. In the RL reward function (Eq. (5)), the balance weight λ and tolerance ξ are set to 0.85 and 2, respectively. For the GRPO objective, the KL regularization weight is $\beta' = 0.01$. This configuration proved stable and effective throughout all experiments.

F Complementary Experiments

F.1 Significance Analysis

Bootstrap-based significance tests confirm that our RL-DoublyCal + Self-Probing significantly outperforms the strongest competitor (SubgraphRAG + Vanilla) on most metrics ($p < 0.05$), whereas Recall on WebQSP is marginally significant, and only Hits on WebQSP and F1 on CWQ do not reach statistical significance. Crucially, DoublyCal achieves an order-of-magnitude reduction in ECE and ACE across all settings, validating its core strength in calibration. The accompanying accuracy gains are a welcome byproduct of improved reliability rather than our primary goal. The ablation study (Table 2) clarifies the underlying mechanism: compared with its SingleCal variant that shares the same proxy and evidence but omits evidence-confidence verbalization, DoublyCal yields comparable F1 while reducing ECE from $\sim$20 to $\sim$4. This demonstrates that evidence confidence indirectly improves reasoning accuracy by encouraging the proxy to prioritize high-confidence evidence, and directly enhances calibration by informing the black-box LLM.

F.2 Evaluation on the MetaQA Datasets

Table 7 reports the results on MetaQA for **DoublyCal**, its ablated variant **SingleCal** (which omits evidence confidence), and the **LLM Reasoner** without knowledge augmentation. All experiments use GPT-3.5-turbo as the black-box LLM with the Vanilla UQ method.

DoublyCal exhibits robustness across KG schemas. Consistent with its competitive performance on WebQSP and CWQ, our RL-DoublyCal attains leading predictive performance across all three MetaQA splits while achieving substantial calibration improvements (e.g., F1 $>$ 85 and ECE $<$ 7). This confirms that the theoretically grounded Bayesian confidence estimator, combined with the generative capacity of an effectively trained LM-based proxy model, constitutes a reliable reasoning solution that generalizes effectively across distinct KGs.

The advantage of DoublyCal becomes more pronounced on harder questions. On 1-hop and 2-hop questions, SingleCal matches DoublyCal in predictive performance yet exhibits markedly inferior calibration. Moreover, both the LLM Reasoner and SingleCal undergo sharp performance declines as question difficulty increases, whereas DoublyCal remains substantially more robust. On the most challenging 3-hop split, RL-DoublyCal attains F1 = 85.4 and ECE = 5.9, surpassing the strongest RL-SingleCal baseline by 5.1 F1 points and 51.2 ECE points. Notably, DoublyCal's low calibration error on hard questions demonstrates its effectiveness in mitigating the overconfidence problem typical of LLMs.

Method	Hits	Recall	F1	ECE ↓	ACE ↓
MetaQA-1hop					
LLM-Reasoner	62.6	54.8	37.4	34.4	34.8
SFT-SingleCal	97.5	95.9	91.6	12.9	12.6
SFT-DoublyCal	97.7	96.0	92.1	4.0	3.7
RL-SingleCal	99.3	98.9	95.3	16.4	16.9
RL-DoublyCal	98.4	98.1	95.1	6.7	5.9
MetaQA-2hop					
LLM-Reasoner	34.9	22.1	15.7	43.4	43.2
SFT-SingleCal	97.9	96.6	94.7	39.8	41.8
SFT-DoublyCal	97.6	96.7	96.1	3.9	3.7
RL-SingleCal	99.5	98.5	97.7	43.7	44.1
RL-DoublyCal	99.3	98.1	97.9	1.2	1.2
MetaQA-3hop					
LLM-Reasoner	51.6	19.8	19.3	28.3	28.3
SFT-SingleCal	94.5	80.6	74.2	50.4	54.2
SFT-DoublyCal	94.8	84.7	78.8	4.9	3.9
RL-SingleCal	95.6	85.0	80.3	57.2	61.7
RL-DoublyCal	96.9	90.4	85.4	5.9	5.7

Table 7: Results (%) on the MetaQA dataset. Light-blue cells in a DoublyCal row indicate improvement over the corresponding SingleCal variant.

F.3 Sensitivity Analysis of the Proxy Base Model

We examine the sensitivity of DoublyCal to the scale of the proxy model by replacing the default Llama2-7B-Chat backbone with two substantially smaller language models:

Qwen3-4B and Qwen3-1.7B [Team, 2025]. All proxy models are trained with SFT followed by RL, and evaluated with the Vanilla UQ method on WebQSP. As reported in Table 8, Llama2-7B exhibits a marginal advantage in predictive metrics (e.g., +1.3 Hits over Qwen3-4B), yet the overall performance of the three proxy backbones remains closely matched across both accuracy and calibration dimensions. These results demonstrate that DoublyCal is robust to significant reductions in proxy model capacity, and it can deliver strong KGQA accuracy together with reliable confidence estimates even under tight computational constraints.

Base Model	Hits	Recall	F1	ECE ↓	ACE ↓
Llama2-7B	91.5	84.8	76.7	4.6	5.1
Qwen3-4B	90.2	83.7	75.3	5.7	5.5
Qwen3-1.7B	90.2	84.2	76.3	4.6	4.8

Table 8: Performance of DoublyCal + Vanilla with different proxy base models on WebQSP.

F.4 Empirical Analysis of Prior Selection

Building upon the theoretical justification in Appendix A, we provide an empirical comparison of different prior choices for Bayesian evidence estimation and analyze their impact on the overall performance of our DoublyCal framework.

The results in Table 9 demonstrate the trade-off between accuracy and calibration controlled by the prior. Empirically, the use of MLE (equivalent to the Haldane prior $\mathrm{Beta}(0,0)$ [Haldane, 1932]) leads to severe calibration degradation, yielding the highest F1 score (75.3) but also the worst ECE (13.4). This aligns with the theoretical risk that point estimates in small-sample extremes produce overconfident outputs, undermining system reliability. While the Uniform prior $\mathrm{Beta}(1,1)$ [Bayes, 1958] applies stronger shrinkage, the Jeffreys prior $\mathrm{Beta}(0.5,0.5)$ [Jeffreys, 1998] provides a better balance, achieving by far the best calibration (ECE 3.1) while maintaining a competitive F1 score (72.6).

Prior Variant	α	β	Hits	Recall	F1	ECE ↓
MLE	0.0	0.0	91.0	81.4	75.3	13.4
Jeffreys prior	0.5	0.5	90.0	81.0	72.6	3.1
Uniform prior	1.0	1.0	89.6	81.8	74.0	6.8

Table 9: Performance (%) of SFT-DoublyCal+Vanilla with different priors for Bayesian estimation on WebQSP.

F.5 Impact of UQ on Predictive Accuracy

To complement the main experiments, we investigate whether prompting the LLM to verbalize its uncertainty affects predictive accuracy in KGQA. We compare recent KGQA methods with our strongest baselines and DoublyCal without explicit uncertainty quantification (UQ).

Table 10 reveals a contrasting pattern. For standard KG-RAG baselines (RoG and SubgraphRAG), verbalized UQ consistently improves F1 scores. This supports the view that uncertainty prompts can induce more deliberate reasoning, which is beneficial for complex questions. In contrast, our RL-DoublyCal already achieves top-tier accuracy without any UQ prompt. Notably, adding UQ with RL-DoublyCal brings no substantial gain and can even slightly lower F1. A plausible explanation is that the high quality of the KG evidence selected by DoublyCal's proxy lowers the intrinsic reasoning difficulty for the primary LLM, thereby reducing the marginal benefit of an extra "second thought" prompted by UQ. This inherent strength positions DoublyCal to better balance the dual objectives of high accuracy and reliable calibration when UQ is employed, as demonstrated by its consistently low ECE in the main results (Table 1).

Reasoning Method	WebQSP	CWQ
SR+NSM [Zhang *et al.*, 2022]	64.1	47.1
SR+NSM+E2E [Zhang *et al.*, 2022]	64.1	46.3
UniKGQA [Jiang *et al.*, 2023]	72.2	49.0
G-Retriever [He *et al.*, 2024]	73.5	-
GNN-RAG [Mavromatis and Karypis, 2025]	71.3	59.4
RoG (GPT-3.5-turbo)	66.8	46.5
+UQ (Self-Probing)	73.5	48.7
SubgraphRAG (GPT-3.5-turbo)	74.7	52.1
+UQ (Vanilla)	77.3	52.2
RL-DoublyCal (GPT-3.5-turbo)	79.7	52.1
+UQ (Self-Probing)	79.3	53.0

Table 10: F1 scores (%) of reasoning methods without and with UQ.

F.6 Case Studies

This appendix provides an extended analysis of the cases presented in Sec. 5.6.

Comparison with Baselines. Table 11 compares RL-DoublyCal with the strongest baselines on the question "Where did George W. Bush live as a child?". None of the methods retrieves an exact supporting fact, because the KG lacks the explicit relation "lived as a child". All methods do retrieve the related fact "George W. Bush was born in New Haven". However, whereas DoublyCal presents concise factual paths accompanied by calibrated confidence scores, the evidence retrieved by RoG and SubgraphRAG is more scattered. Consequently, the LLMs guided by RoG and SubgraphRAG are distracted from the core entities ("George W. Bush" and "New Haven") and assign overconfident scores (0.9–1.0) to the incorrect prediction "Midland".

In contrast, DoublyCal attaches calibrated confidence scores through its first-stage calibration. Specifically, the precise path about birthplace receives a high score (0.8), while the more generic expansions receive lower scores (0.5), reflecting their weaker inferential relevance. Guided by these scores, the black-box LLM correctly assigns low confidence (0.3) to the plausible but incorrect prediction "Connecticut", demonstrating better-calibrated uncertainty estimation.

Comparison with Ablated Models. Table 12 contrasts the full DoublyCal framework with its SingleCal ablation, which removes the calibrated evidence confidence (i.e.,

Sample	
Question:	Where did George W. Bush live as a child?
Answers:	New Haven.
RL-DoublyCal + Self-Probing	
Retrieval:	George W. Bush → people.person.place_of_birth → New Haven [Confidence: 0.8] George W. Bush → people.person.place_of_birth → New Haven → location.location.containedby → Connecticut [Confidence: 0.5] George W. Bush → people.person.place_of_birth → New Haven → location.location.containedby → United States of America [Confidence: 0.5]
Predictions:	{Connecticut: 0.3}
RoG + Self-Probing	
Retrieval:	George W. Bush → people.person.place_of_birth → New Haven George W. Bush → people.place_lived.person → m.03prwzr → people.place_lived.location → Midland George W. Bush → people.person.nationality → United States of America → location.location.containedby → St. Louis ...
Predictions:	{Midland: 0.9}
SubgraphRAG + Vanilla	
Retrieval:	(George W. Bush, people.person.place_of_birth, New Haven) (George W. Bush, people.person.nationality, United States of America) ... (m.03prwzr, people.place_lived.location, Midland) (m.02xlp0j, people.place_lived.location, Washington, D.C.) ...
Predictions:	{Midland: 1.0}

Table 11: Comparative case study: DoublyCal vs. baselines on calibration.

Sample	
Question:	Where was Martin Luther King, Jr. raised?
Answers:	Atlanta.
RL-DoublyCal (Full) + Vanilla	
Retrieval:	Martin Luther King, Jr. → people.person.place_of_birth → Atlanta [Confidence: 0.8] Martin Luther King, Jr. → people.deceased_person.place_of_death → Memphis [Confidence: 0.8]
Predictions:	{Atlanta: 0.8, Memphis: 0.1}
RL-DoublyCal (SingleCal) + Vanilla	
Retrieval:	Martin Luther King, Jr. → people.person.place_of_birth → Atlanta Martin Luther King, Jr. → people.deceased_person.place_of_death → Memphis
Predictions:	{Atlanta: 0.8, Memphis: 0.2}
SFT-DoublyCal (Full) + Vanilla	
Retrieval:	Martin Luther King, Jr. → people.person.place_of_birth → Atlanta [Confidence: 0.8] Martin Luther King, Jr. → people.person.nationality → United States of America → location.country.capital → Washington, D.C. [Confidence: 0.8]
Predictions:	{Atlanta: 0.8, United States of America: 0.2}
SFT-DoublyCal (SingleCal) + Vanilla	
Retrieval:	Martin Luther King, Jr. → people.person.place_of_birth → Atlanta Martin Luther King, Jr. → people.person.nationality → United States of America → location.country.capital → Washington, D.C.]
Predictions:	{Atlanta: 0.7, United States of America: 0.3}

Table 12: Ablation case study: Full vs. SingleCal variant.

only the second-stage calibration remains). Both RL- and SFT-DoublyCal retrieve high-confidence evidence focused on the question (e.g., "Martin Luther King, Jr. was born in Atlanta"). RL-DoublyCal exhibits slightly sharper calibration, likely because its reward-driven training promotes more discriminative evidence selection. More importantly, when evidence confidence is provided (the full model), the LLM's predicted confidence is more concentrated on the correct answer. For example, RL-DoublyCal (Full) assigns only 0.1 confidence to the distracting alternative "Memphis", whereas its SingleCal variant assigns 0.2. Similarly, SFT-DoublyCal (Full) assigns 0.2 to "United States of America", while the SingleCal variant assigns 0.3. This directly demonstrates that the first-stage evidence calibration is crucial for providing a reliable confidence anchor, enabling the LLM to synthesize multiple evidence pieces into a decisive and well-calibrated final prediction.