Overthinking as a Symptom of Knowledge Conflict: Understanding and Detecting LLM's Hallucinations in Retrieval-Augmented Question Answering

Zhihua Wen, Li Hao[1], Zhiliang Tian, Zhizhao Liu, Yanqi Shi,
Yiping Song[*], Dongsheng Li[*]
National University of Defense Technology
{zhwen, lihao_, tianzhiliang, liuzhizhao,}@nudt.edu.cn,
{songyiping, dsli}@nudt.edu.cn,

Abstract

Retrieval-augmented generation (RAG) is increasingly used to ground Large language Models (LLMs) with up-to-date or private information, and has become a key component for knowledge-intensive LLM systems (e.g., LLM agents). However, RAG with LLMs often hallucinate due to unreliable retrieved information or outdated internal knowledge, making hallucination detection crucial for reliable LLM systems. Recently, reasoning-intensive LLMs that produce explicit reasoning trajectories before answering have shown promise in RAG by examining and synthesizing retrieved evidence. Yet, they often suffer from `overthinking`: generating redundant or unproductive reasoning with limited performance gains. While prior work mainly studies overthinking in closed-book reasoning tasks that mainly rely on internal knowledge, overthinking in RAG-based question answering (QA) that requires open-book reasoning over both internal knowledge and external context remains underexplored. Thereby, we conduct empirical studies to investigate the relationship between overthinking and hallucination in RAG-based QA and find: (1) Overthinking in reasoning LLMs produces lengthy reasoning trajectories that are correlated with hallucinations; (2) This overthinking is often associated with knowledge conflict, where LLMs struggle to reconcile retrieved context with internal knowledge. Based on our findings, we propose a knowledge conflict-adaptive hallucination detection framework. We train a classifier to identify the three knowledge conflict reasoning patterns, design a knowledge conflict score to measure knowledge conflict severity, and apply tailored detection strategies with different prompts for different conflict levels. Unlike existing methods, our approach requires neither LLM parameter access nor retrieval reliability, and only needs the target LLM output from a single generation run. Experiments on three QA datasets and three LLMs confirm our framework's effectiveness.

1 Introduction

Large language models (LLMs) are increasingly used in knowledge-intensive LLM workflows (e.g., agents), where models must retrieve, synthesize, and reason over external evidence before producing answers. Retrieval-augmented generation (RAG) has thus become a basic mechanism for connecting static parametric LLMs with current, domain-specific, or private knowledge, and serves as a key component in agentic systems involving multi-step reasoning [1, 2]. However, RAG with LLM often suffers from hallucinations, producing untrustworthy content. Retrieved evidence can be incomplete,

[*]Correponding Authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

noisy, or mutually inconsistent, while the LLM's parametric knowledge may also be outdated. In such settings, detecting hallucinations is critical to the reliability of LLM-generated answers in knowledge-intensive LLM systems.

Recently, reasoning-intensive LLMs that generate human-like, explicit reasoning trajectories before producing responses (e.g., o1 [3] and Deepseek-R1 [4]) have shown strong abilities on complex reasoning tasks. In RAG, such models are appealing because their step-by-step reasoning may help validate retrieved evidence and synthesize information from multiple sources. However, recent studies reveal that reasoning LLMs often suffer from `overthinking`: they may produce overly redundant or unproductive reasoning steps that bring limited or even negative performance gains [5–7]. Existing studies of overthinking mainly focus on closed-book reasoning tasks [8, 9], where the model primarily relies on internal parametric knowledge and follows established rules[2]. In contrast, overthinking in open-book reasoning tasks (e.g., RAG-based multihop QA) remains less explored. This setting is fundamentally different because the model must not only reason over the question, but also decide how to use retrieved evidence together with its own internal knowledge [10–12].

To investigate the overthinking issue and its relation to hallucination in RAG-based QA, we conduct an empirical study on multiple multihop QA datasets and arrive at two key findings. **First**, reasoning LLMs in RAG often overthink, producing lengthy reasoning is correlated with incorrect answers (hallucinations). **Second**, this overthinking behavior in RAG is often associated with *knowledge conflict*, where the LLM struggles to reconcile information from the retrieved context with its internal parametric knowledge. We identify three main reasoning patterns, each manifesting as a distinct pattern in the model's reasoning trajectory: 1) *Instruction Reinterpretation*: LLM subtly shifts, reinterprets, or attempts to rationalize its modifications to the user's original instructions (e.g., "Assuming that the user has made a mistake ..."). 2) *Context Insufficiency Alarming*: LLM expresses that the retrieved context lacks sufficient information to answer the question. 3) *Explicit Internal Reasoning*: LLM explicitly states that it is considering its internal knowledge, often through linguistic markers that distinguish this knowledge from the retrieved context. For example, LLM may use phrases like "according to general knowledge," which signal a verbal acknowledgement of relying on information beyond the retrieved context.

Based on our findings, we propose ConfliDetect, a knowledge conflict-adaptive hallucination detection framework for reasoning LLMs in RAG-based QA. First, we train a classifier to identify the three knowledge conflict reasoning patterns. Second, we design a knowledge conflict score to quantify the severity of LLM overthinking due to knowledge conflict. Based on this score, we categorize LLM reasoning into three types: no conflict, mild conflict, and severe conflict. For each category, we design distinct strategies via prompting-based detection tailored to the severity of knowledge conflict, guiding a detector LLM to assess hallucination. Besides, we design a repetitive reasoning reduction strategy to simplify the reasoning trajectory and improve detection efficiency. Our method explores a new perspective to detect hallucinations in RAG-based QA: enabling monitoring of warning reasoning steps and detecting hallucination based on knowledge conflict. It requires fewer data permissions than existing black-box [13] or white-box [14, 15] hallucination detection approaches and can detect hallucination using only the target LLM output from a single generation run. It requires no access to LLM parameters, nor does it depend on the assumption of retrieval reliability (i.e., assuming that the retrieved information is always reliable). This makes our approach more resilient in real-world scenarios where retrieved information is often noisy or incomplete.

Our contributions are as follows: (1) To our knowledge, our work is one of the first to explore the overthinking issue and its association with hallucination in RAG-based QA. (2) We uncover a consistent empirical association between LLM overthinking and hallucination in RAG, and identify three knowledge conflict reasoning patterns as warning signals of LLM hallucination. (3) We propose a plug-and-play framework for hallucination detection in RAG-based QA, requiring only the target LLM output from a single generation run. (4) Extensive experiments across multiple LLMs and QA datasets validate our findings and the effectiveness of our framework.

[2]For example, applying mathematical theorems to solve math problems or following programming-language syntax in code generation.

2 Related Work

2.1 Overthinking in Reasoning-intensive LLMs

Reasoning LLMs demonstrate impressive capabilities through extended deep reasoning before providing an answer [16], but they often suffer from "overthinking". Overthinking occurs when reasoning LLMs expend excessively long reasoning trajectories, with minimal performance gains [17, 18]. Current studies explore overthinking in closed-book reasoning tasks where LLMs mainly leverage internal knowledge to reason based on established rules (e.g., following mathematical theorems in math tasks [5, 8] or the syntax and grammar in code generation [19]). Fan et al. (2025) found that LLMs cannot identify ill-posed, unanswerable math questions and tend to overthink due to a lack of critical thinking. Zhang et al. (2025) discovered that, in mathematical tasks, even though the LLMs' internal parameters can already predict the correct answer, they still cannot stop generating redundant reasoning steps. Yang et al. (2025) argued that overthinking stems from LLM's inability to internalize the search process. Current research rarely explores open-book reasoning tasks that require both internal parametric knowledge and external facts for reasoning, which may exhibit underexplored LLM reasoning behaviors.

2.2 LLM Hallucination in RAG

Hallucination in LLMs refers to generating factually incorrect or fabricated information [21]. While RAG generally enhances truthfulness in many tasks [10, 22, 23], hallucinations still arise from the complex interaction between LLM's internal knowledge and external retrieval contexts [24, 25]. On one hand, LLMs may prioritize contextual information over their internal parametric knowledge, even when the context is incorrect [26, 27] or outdated [28]. On the other hand, LLM's internal biases or misconceptions can lead to the misinterpretation of input [29, 30]. Recent studies suggests that LLMs' preference for dominant knowledge often overshadows long-tail knowledge [31], complicating the detection of hallucination.

Existing studies explore various hallucination detection methods. For analysis of LLM internal states, researchers leverage LLM hidden states [14], activation-based accuracy prediction [15, 32], attention maps [33], or contrastive token probabilities like REFIND [34]. Uncertainty-based approaches evaluate generation stability through semantic entropy [35] or cross-examination via multiple responses [36, 37]. To ensure grounding in RAG, specialized methods check consistency against external context using claim verification [38, 39], metamorphic testing [40], or NLI-based aggregation [41].

Despite these advancements, most methods train specialized detector models, require access to LLM parameters or assume that retrieved external contexts are always reliable. While reasoning LLMs provide explicit and understandable reasoning trajectories that expose the LLM's internal logic, leveraging such behaviors for hallucination detection remains relatively underexplored.

3 Empirical Study

We study LLM overthinking in RAG-based QA and its relationship with hallucination. Via semantic-level reasoning analysis, we identify distinct reasoning behaviors and unveil how overthinking correlates with hallucinations.

3.1 Experimental setting

We evaluate Deepseek-R1[3] [4], a large-scale (671B parameters) mixture of experts (MoE) model, and GLM-Z1 [43], dense Transformer architecture LLM, to investigate performance on a smaller scale (9B parameters), representing two mainstream LLM architectures [44]. We conduct experiments on widely adopted QA datasets for RAG: TempQA [45], MuSiQue [46], and BamBoogle [47] (see detailed dataset descriptions in App P). Following widely recognized settings [48], we construct the retrieval repository using the Wiki19 snapshot with widely adopted bge-large-en [49] as the retriever. Each dataset $\mathcal{Q}$ comprises multiple samples, each with a query q and a gold answer a_g, where

[3]Due to OpenAI's restriction [42] on user access to the reasoning trajectories in their reasoning models (o1 and o3), we cannot conduct experiments on their products.

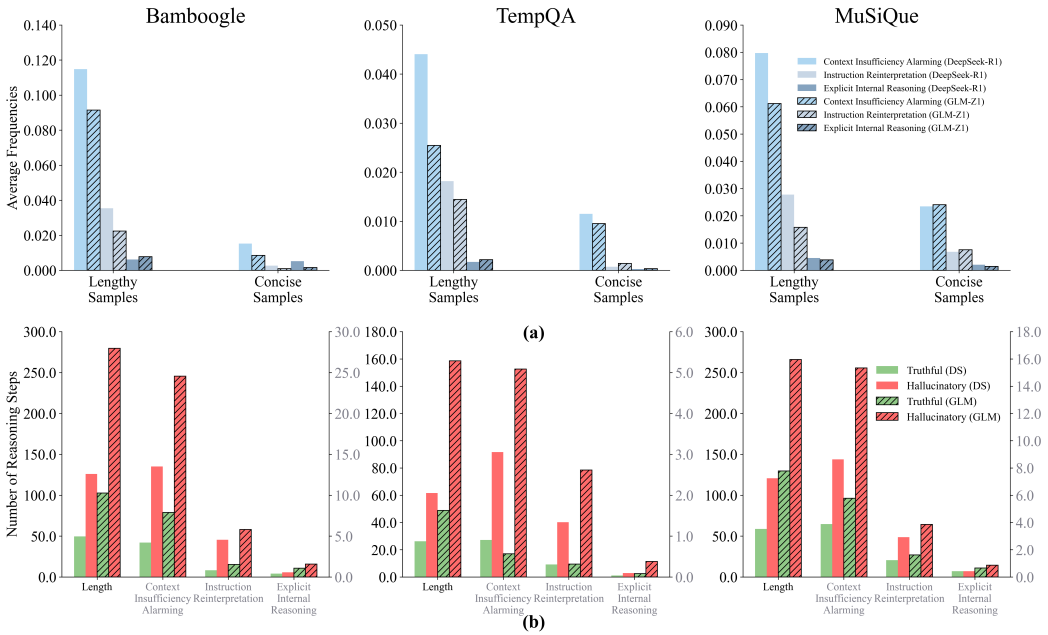

Figure 1: Results of our empirical studies. Charts in **(a)** show the average frequencies of three characteristic reasoning patterns by comparing *Lengthy Samples* (reasoning steps $>$ mean) and *Concise Samples* ($\leq$ mean). Solid bars represent DeepSeek-R1 and hatched bars represent GLM-Z1. Charts in **(b)** compare *Truthful* (green) and *Hallucinatory* (red) samples regarding their average trajectory length (left y-axis) and the average occurrence of the three reasoning patterns (right y-axis).

answering q requires multi-hop reasoning. For each query q, we retrieve the top 10 relevant passages and feed them into the reasoning LLM, along with the query. The model generates a reasoning trajectory $R = \{r_1, \ldots, r_N\}$, where each r_i is a sentence representing an individual reasoning step[4], before producing its response a. Finally, we use Deepseek-V3 [50] to check the consistency between a and the ground truth answer a_g to evaluate whether the LLM produces a hallucinatory (i.e., incorrect) answer or a truthful (i.e., correct) answer (see prompt template in App. S). We report Recall@10 and stratify detection results by whether the gold answer appears in the retrieved top-10 passages; these analyses characterize retrieval quality without assuming that every retrieved passage is correct.

3.2 How Does LLM Overthinking Correlate with Hallucination in RAG-based QA?

We conduct an empirical study to investigate the characteristics of reasoning trajectories and their correlation with hallucinations. Through manual observation, we identify three characteristic reasoning patterns that frequently emerge during extensive reasoning: (1) *Instruction Reinterpretation (IR)*: The model persistently reinterprets, shifts, or attempts to rationalize modifications to the user's original instructions (e.g., "Assuming that the user has made a mistake ..., the answer could be correct." and "Maybe the user confused ..."). (2) *Context Insufficiency Alarming (CIA)*: The model repeatedly expresses that the retrieved context lacks sufficient detail or relevance (e.g., "is not clear" and "the details about ... is missing"). (3) *Explicit Internal Reasoning (EIR)*: The model explicitly invokes its internal knowledge during reasoning (e.g., "according to real-world knowledge" and "based on my understanding").

Specifically, to detect these patterns, we construct three sets of semantically related text spans corresponding to the above types (see text spans in App. R). We then examine each step r in a trajectory to determine if it contains at least one of the text spans from these sets. If a step contains a span, it is classified as exhibiting the corresponding reasoning pattern. To analyze the relationship between overthinking and these patterns, we first split each dataset into *Lengthy Samples* (reasoning steps $>$ mean) and *Concise Samples* ($\leq$ mean). For these groups, we compute the average frequency

[4]We use the NLTK library to divide reasoning step.

of each pattern (calculated as the number of pattern occurrences divided by the total reasoning steps in each sample). Subsequently, to explore the link between these behaviors and model reliability, we split the dataset into *Truthful* and *Hallucinatory* subsets based on response correctness. For these groups, we compute the average reasoning length and the average number of occurrences of each pattern. See results in Fig. 1. Here, overthinking is a trajectory-level description of inefficient or redundant reasoning, rather than an independently supervised prediction target. ConfliDetect operationalizes overthinking-related inefficiency through step-level knowledge-conflict signals that frequently co-occur with harmful overthinking. A long trajectory with few conflict signals may reflect productive multi-hop verification, whereas frequent conflict signals are treated as warning indicators.

Observation 1: Lengthy samples are characterized by a higher prevalence of specific reasoning patterns. As shown in Fig. 1(a), across all settings, *Lengthy Samples* consistently exhibit a higher frequency of all three reasoning patterns compared to *Concise Samples*. Notably, the relative proportions of these patterns follow a remarkably similar trend across different architectures (solid vs. hatched bars). This indicates that the increased length in reasoning trajectories is consistently associated with the emergence of these specific reasoning behaviors. We characterize this state as the LLM experiencing *knowledge conflict*, which indicates that the LLM struggles to reconcile external information with its internal knowledge.

Observation 2: Hallucinations are strongly correlated with both reasoning length and the occurrence of these patterns. As shown in Fig. 1(b), hallucinatory samples (red bars) consistently demonstrate significantly longer reasoning trajectories and a higher frequency of the three reasoning patterns than truthful samples (green bars). The fact that all indicators of knowledge conflict are higher in hallucinatory samples suggests a strong positive correlation between knowledge conflict and hallucination. This pattern is consistent with higher hallucination risk when the model struggles to reconcile the two knowledge sources, while the observational analysis does not identify a causal mechanism.

Observation 3: Overthinking is most severe when retrieval conflicts with correct internal knowledge, while hallucination remains high whenever either knowledge source is unreliable.

To further validate that this correlation is driven by RAG-specific knowledge conflict rather than general model uncertainty, we disentangle the reliability of the retrieved context and the LLM's internal parametric knowledge. Specifically, we label the retrieved context as reliable if it contains the ground-truth answer, and label the internal knowledge as reliable if the same LLM can answer the question correctly in a closed-book setting without retrieval. We categorize samples into four fine-grained scenarios: *Both Reliable* (correct retrieval and correct internal knowledge), *Wrong Retrieval Only* (incorrect retrieval but correct internal knowledge), *Wrong Internal Knowledge Only* (correct retrieval but incorrect internal knowledge), and *Both Unreliable* (both sources are incorrect). As shown in Tab. 1, we observe three key findings:

Table 1: Statistics of hallucination rate, average reasoning length, and conflict pattern frequencies across different knowledge reliability categories.

Category	Prop.	Hall. Rate	Avg. Len.	CIA %	IR %	EIR %
Both Reliable	30.0%	21.6%	3,176	1.83	0.39	0.14
Wrong Retrieval Only	15.5%	55.8%	8,671	6.13	2.19	0.54
Wrong Internal Only	7.7%	82.4%	4,916	2.63	0.91	0.13
Both Unreliable	46.9%	85.2%	8,317	4.87	1.57	0.24

(1) *Both Reliable* achieves the lowest hallucination rate (21.6%), the shortest reasoning length (3,176), and the lowest conflict frequencies. Missing either knowledge source is associated with significant performance degradation. (2) *Wrong Retrieval Only* produces the longest reasoning (8,671) and the highest conflict frequencies. Incorrect retrieved content co-occurs with disagreement with the model's correct internal knowledge, and these samples contain more repeated attempts to reconcile the contradiction. (3) *Wrong Internal Knowledge Only* has the second-highest hallucination rate (82.4%) but a relatively shorter reasoning length (4,916). Although correct evidence is present in the retrieved context, the LLM may fail to identify or properly use it due to incorrect parametric knowledge. Thus, evidence availability does not guarantee evidence utilization. Since the LLM

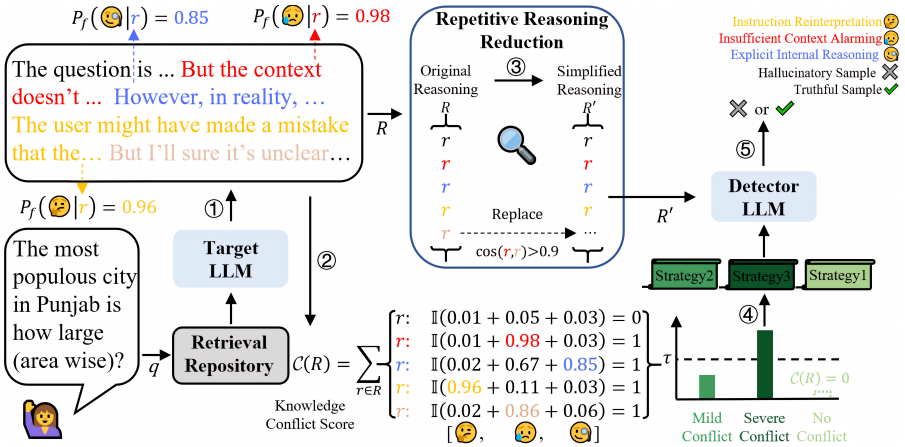

Figure 2: Overview of our ConfliDetect framework. ① The target LLM generates reasoning trajectory R for query q in RAG. ② Knowledge conflict classifier f examines each r in R to calculate the knowledge conflict score $\mathcal{C}(R)$. ③ Repetitive reasoning r is replaced by placeholder "…" to obtain simplified R'. ④ R is categorized as Severe Conflict since $\mathcal{C}(R)$ exceeds threshold τ. ⑤ The detector LLM applies adaptive strategies based on conflict severity to output the final result (✓ or ×).

lacks a correct internal anchor to conflict with the retrieved context, it exhibits less overthinking than *Wrong Retrieval Only* while still being prone to hallucination.

Our findings are applicable to real-world RAG-based QA applications (e.g., search engines), where untrustworthy external context or outdated internal knowledge is associated with knowledge conflict, longer reasoning, and hallucinations. App H provides additional analysis showing that our findings capture the vast majority of hallucination-related conflict behaviors and specifically target abnormal reasoning signals beyond ordinary reasoning steps. Sec. K further verify our findings on a more realistic setting.

4 ConfliDetect: A Knowledge Conflict-Adaptive Hallucination Detection Framework

The overview of our framework is in Fig. 2. We first introduce the knowledge conflict score, a metric to evaluate the level of knowledge conflict exhibited in the reasoning trajectory in RAG-based QA. Then, based on this score, we categorize samples into three levels of knowledge conflict severity and apply different detection strategies.

4.1 Knowledge Conflict Score: Measuring Reasoning-time Knowledge Conflict for RAG-based QA

We introduce the knowledge conflict score, a metric quantifying the severity of knowledge conflict in reasoning by detecting steps that exhibit knowledge conflict. To achieve this, we build a dataset to train a conflict classifier.

Dataset Construction. We construct a dataset to train a knowledge conflict classifier via LLM-as-judge annotation (existing studies verify the effectiveness of LLM annotation [51]). For question q, we collect an LLM's all reasoning steps R under RAG. Then, we design prompt template for each type of conflict, and use GPT4o to annotate each $r \in R$ accordingly. We validate the reliability of LLM-based data annotation by comparing kappa agreement scores with human annotations in App. C. See dataset details in App. D.1.

Model Training. We fine-tune a multi-class knowledge conflict classifier based on DeBERTa-v3-base [5]. The classifier $f()$ takes a reasoning step r, and outputs probabilities of three types of knowledge conflict reasoning. The training objective is to minimize the binary cross-entropy loss for

[5] https://huggingface.co/microsoft/deberta-v3-base

multi-class classification:

$$\mathcal{L}(\mathbf{y}, f(r)) = -\sum_{k=1}^{C} \Big[y_k \log \sigma(p_k) + (1 - y_k) \log(1 - \sigma(p_k)) \Big] \tag{1}$$

where $C = 3$ denotes the number of knowledge conflict reasoning types, $\mathbf{y} = [y_1, \ldots, y_C]$ is the label vector with $y_k \in \{0, 1\}$ indicating whether the reasoning step exhibits the k-th reasoning type. $f(r) = [p_1, \ldots, p_C]$ are the logits produced by the classification layer for each type. Notice that a single reasoning step may simultaneously exhibit multiple types of conflict. See analysis of robustness and advantages of our deberta-based classifier in App. E, training details in App. D.2.

Building on the detection of each reasoning step, we introduce a knowledge conflict score to measure the overall severity of knowledge conflict in RAG reasoning. The key insight is that knowledge conflicts manifest at the step level during reasoning, and the cumulative occurrence of such conflicts is associated with the reliability of the reasoning trajectory. As in Eq. 2, $\hat{\mathbf{y}}^{(r)} = [\hat{y}_1^{(r)}, \ldots, \hat{y}_C^{(r)}]$ denotes the prediction vector ($\hat{y}_k \in \{0, 1\}$) for reasoning step r. The knowledge conflict score for a trajectory R is computed as:

$$\mathcal{C}(R) = \sum_{r \in R} \mathbb{I}\left(\sum_{k=1}^{C} \hat{y}_k^{(r)} \geq 1 \right), \tag{2}$$

where $\mathbb{I}(\cdot)$ is 1 if the condition is true, 0 otherwise. $\mathcal{C}(R)$ counts the number of reasoning steps that exhibit at least one type of conflict, providing a simple conflict-related reliability signal. The rationale is that each conflict step is associated with confusion that may be correlated with errors through the reasoning chain, making the count of such steps a meaningful indicator of overall quality. Sec. G analyze $\mathcal{C}(R)$'s robustness.

4.2 Adaptive Hallucination Detection Based on Knowledge Conflict Severity

We detect hallucinations adaptively by tailoring evaluation strategies to the severity of knowledge conflict observed in the LLM's reasoning trajectory. Specifically, for each sample, we first compute its knowledge conflict score $\mathcal{C}(R)$ using our trained classifier. Based on a threshold τ, we categorize each sample into three distinct levels: (1) No Knowledge Conflict if $\mathcal{C}(R) = 0$, (2) Mild Knowledge Conflict if $0 < \mathcal{C}(R) \leq \tau$, and (3) Severe Knowledge Conflict if $\mathcal{C}(R) > \tau$.

Next, we design three prompt-based strategies for different categories (details in App. S) and employ a detector LLM to detect hallucinations. Intuitively, our design logic is: a clear and sound reasoning process may only needs a quick check, but a conflicted one needs a more rigorous inspection. For no conflict, the detector verifies the logical rigor and relevance of each reasoning step while checking for implicit flaws or misunderstandings. For mild conflict, the detector assesses whether the minor conflicts affect the answer correctness. For severe conflict, the detector determines whether the answer is derived from a more reliable information source by distinguishing between reasoning based on internal knowledge versus external context. App. F shows a positive correlation between hallucination and conflict severity, supporting the use of conflict severity as a reliable signal for adapting verification rigor.

Before detection, we notice that many samples contain repetitive reasoning steps, which may confuse the detector LLM and make the output more uncertain. To mitigate, we propose a repetitive reasoning reduction algorithm to filter repetitive reasoning steps. The algorithm (see pseudocode in Algorithm 1) employs an embedding model E to encode each reasoning step and iteratively identify redundancies. For each step r_i in R, we first encode it; if it has not been marked as repetitive, we add it to the output list R'. We then compare r_i with all subsequent steps by computing cosine similarity, and if the similarity exceeds 0.9, we mark the repetitive steps and replace them with a placeholder token ("..."). We set this threshold following common practice in near-duplicate detection with sentence embeddings to identify semantically redundant text while minimizing false positives [52]. This process continues for all steps, resulting in a filtered reasoning trajectory where only semantically unique steps are retained. By removing repetitive steps, the algorithm reduces noise and prevents the detector LLM from being overwhelmed by lengthy contexts. The final filtered steps R' are then used as input for the detector. App. L analyzes the time complexity of ConfliDetect and shows that

Table 2: Performance of different hallucination detection methods. Best results are bolded.

Acc/F1	Deepseek-R1 (671B)			GLM-Z1 (9B)			Qwen3 (32B)		
Method	Bamboogle	TempQA	MuSiQue	Bamboogle	TempQA	MuSiQue	Bamboogle	TempQA	MuSiQue
VanillaLLM [55]	0.66/0.59	0.67/0.29	0.63/0.72	0.71/0.70	0.69/0.39	0.70/0.79	0.67/0.62	0.71/0.31	0.66/0.75
SelfCheck [13]	0.65/0.59	0.72/0.44	0.63/0.72	0.68/0.67	0.73/0.31	0.71/0.80	0.66/0.60	0.73/0.31	0.67/0.75
LexiDetect [6]	0.72/0.75	0.67/0.52	0.78/0.86	0.66/0.73	0.68/0.62	0.81/0.89	0.67/0.70	0.69/0.52	0.73/0.83
InterrLLM [37]	0.45/0.43	0.30/0.34	0.48/0.38	0.46/0.52	0.31/0.40	0.49/0.61	0.44/0.61	0.28/0.43	0.70/0.82
SemanticEntropy [56]	0.72/0.71	0.71/0.50	0.69/0.78	0.64/0.63	0.71/0.60	0.55/0.68	0.62/0.58	0.70/0.50	0.64/0.69
ConfliDetect	**0.81/0.81**	**0.72/0.60**	**0.78/0.86**	**0.74/0.77**	**0.75/0.65**	**0.82/0.89**	**0.75/0.74**	**0.75/0.57**	**0.76/0.84**

its computational overhead is low, as the lightweight preprocessing steps help compress reasoning trajectories before final LLM-based detection.

In this way, we monitor LLM reasoning steps and detect hallucinations accordingly. Our method requires fewer data permissions than existing hallucination detection methods. Compared to them, our approach does not require: 1) access to LLM parameters or output probabilities [35, 53]. 2) sample multiple generations to measure consistency [13, 53]. 3) the reliance of retrieval reliability [34, 40]. Our framework explores a new paradigm for hallucination detection. It alerts users upon encountering knowledge conflicts during reasoning and detect hallucinations from a single target generation run.

5 Experiments

5.1 Setting

Implementation Details. In addition to the reasoning LLMs used in the empirical studies, we also evaluate on Qwen3-32b [54]. Following Yehuda et al. (2024), we adopt **Acc** and **F1** to evaluate the hallucination detection performance. We set $\tau = 10$ based on our empirical findings and use bge-large-en [49] as the embedding model E in Sec. 4.2. We employ Deepseek-R1 as the hallucination detector LLM for all baselines[6].

Baseline. We compare our method with the following baselines that align with our framework's black-box and training-free nature (See discussion of our baseline selection in App. Q, prompt templates in App. S). (1) VanillaLLM [55] directly prompts an LLM to evaluate the response correctness. (2) SelfCheck [13] detects hallucinations by voting over five evaluations based on VanillaLLM.(3) Inspired by Fan et al. (2025), LexiDetect incorporates findings on overthinking patterns and lexical indicators (e.g., "wait", "however") to instruct the LLM detector in detecting hallucinations. (4) InterrLLM [37] reconstructs query multiple times based on LLM response and measures the semantic similarities between the original and reconstructed queries. (5) SemanticEntropy [56] estimates uncertainty by sampling multiple responses, grouping semantically equivalent answers, and computing entropy over the resulting semantic clusters to detect hallucinations. Additional controls for trajectory length, conflict-classifier choice, retrieval quality, question difficulty, and trace completeness are reported in App. **??**. On Bamboogle, ConfliDetect achieves 0.81/0.81 Acc/F1, compared with 0.62/0.58 for a length-only detector and 0.75/0.70 under controlled reasoning length, indicating that the conflict patterns provide information beyond trajectory length.

5.2 Overall Performance

We evaluate the effectiveness of our framework by comparing it with multiple baselines across multiple reasoning LLMs and datasets (see Tab. 2). For each sample, we input the query, the LLM reasoning and final answer to predict its hallucination label. VanillaLLM and SelfCheck provide basic detection capabilities but exhibit limited reliability when faced with the complex reasoning paths of multi-hop queries. InterrLLM yields the lowest performance because the reconstruction of multi-hop queries often introduces cumulative errors that mislead the detector. While LexiDetect improves upon basic baselines by monitoring lexical features of overthinking, it remains unable to accurately identify necessary productive reasoning and conflicted reasoning. SemanticEntropy achieves competitive performance in some settings, but its uncertainty signal is not specifically tailored to reasoning-induced knowledge conflicts and therefore can be less stable across settings. Our

[6]The code is included in the supplementary materials and will be publicly released upon acceptance.

Table 3: Ablation study on the key components in ConfliDetect.

Acc/F1	Deepseek-R1 (671B)			GLM-Z1 (9B)			Qwen3 (32B)		
Variants	Bamboogle	TempQA	MuSiQue	Bamboogle	TempQA	MuSiQue	Bamboogle	TempQA	MuSiQue
− *Adaptive Detect*	0.74/0.70	0.71/0.42	0.63/0.71	0.72/0.71	0.74/0.51	0.70/0.79	0.70/0.63	0.75/0.45	0.66/0.74
− *Reduce Redundancy*	0.75/0.76	0.72/0.59	0.78/0.86	0.67/0.73	0.71/0.64	0.80/0.88	0.68/0.67	0.74/0.56	0.75/0.83
− *Detector*	0.73/0.71	0.72/0.52	0.73/0.81	0.70/0.72	0.72/0.62	0.78/0.86	0.65/0.64	0.74/0.55	0.75/0.83
ConfliDetect	**0.81/0.81**	**0.72/0.60**	**0.78/0.86**	**0.74/0.77**	**0.75/0.65**	**0.82/0.89**	**0.75/0.74**	**0.75/0.57**	**0.76/0.84**

method consistently achieves the best performance by explicitly leveraging the robust relationship between knowledge conflict patterns and reasoning-induced hallucinations. The consistent gains across diverse target LLMs like Qwen and GLM demonstrate that our framework is robust and independent of LLM architectures. Statistical significance tests (App. A) confirm the improvements are statistically significant ($p < 0.05$). See case study in App. O. Sec. K further verifies ConfliDetect's effectiveness in a more realistic setting.

5.3 Ablation Study

We conduct an ablation study on our framework to evaluate the contribution of each component in detecting hallucinations (see Tab. 3). − *Adaptive Detect* replaces the adaptive detection that tailors verification prompts based on conflict severity with a uniform verification prompt. Without distinguishing the severity of knowledge conflict, the framework fails to provide targeted guidance for the detector LLM, resulting in suboptimal performance across all settings. − *Reduce Redundancy* abandons the repetitive reasoning reduction strategy and feeds the full reasoning trajectory to the detector LLM. This leads to two main issues: (1) the detector LLM must process lengthy, repetitive reasoning steps, which increases computational overhead and may distract from critical reasoning patterns; (2) repetitive contexts can overwhelm the signal of knowledge conflict. The consistent performance degradation across almost all settings validates the importance of filtering out redundant reasoning. − *Detector* removes the final LLM-based detector and directly uses the knowledge conflict score $\mathcal{C}(\mathcal{R})$ for hallucination prediction, where responses with high conflict scores are classified as hallucinated and those with low scores are classified as correct. − *Detector* consistently underperforms ConfliDetect, indicating that conflict severity should not be directly mapped to hallucination labels. Hallucination can remain high even when such conflict is less severe (see Observation 3 in Sec. 3). Therefore, a detector LLM is necessary to perform fine-grained verification over the reasoning trajectory for hallucination detection. Besides, we confirm that ConfliDetect remains robust to key hyperparameters and design choices, including the conflict threshold τ, reasoning length, and detection prompts, via a fine-grained sensitivity analysis in App. B. App. G further verifies that $\mathcal{C}(\mathcal{R})$ is robust to threshold choice and reasoning length, consistently outperforming a length-normalized variant under threshold-independent metrics.

5.4 Impact of Detector LLM Choice

To assess whether our framework generalizes across different detector LLMs, we replace DeepSeek-R1 with other powerful LLMs, GPT-4o, Claude-Sonnet-4.6 (Claude), and GPT-5. We compare ConfliDetect with the Vanilla baseline, using the same detector LLM without our conflict-adaptive detecting framework.

Tab 4 shows two key observations. First, stronger detector LLMs yield better performance for both the Vanilla baseline and our framework (e.g., GPT-5 achieves the highest scores), which aligns with the intuitive expectation that a more capable LLM serves as a better detector. Second, our framework consistently and significantly outperforms the Vanilla baseline regardless of the chosen detector LLM. For instance, on hallucinations produced by DSR1, augmenting GPT-4o with our conflict-adaptive framework improves Acc from 0.59 to 0.74. The same trend holds for stronger detector LLMs: Claude improves from 0.72 to 0.84, and even GPT-5 improves from 0.85 to 0.94. **These results provide strong evidence that ConfliDetect generalizes across diverse detector LLMs, strengthening advanced LLMs' capability through our conflict-adaptive detection framework.**

Table 4: Performance (Acc/F1) on Bamboogle using different detector LLMs. ConfliDetect consistently outperforms the Vanilla baseline regardless of the detector choice.

Detector Setting	DSR1		GLM-Z1		Qwen3	
	Acc	F1	Acc	F1	Acc	F1
GPT-4o (Vanilla)	0.59	0.65	0.63	0.70	0.66	0.71
GPT-4o (ConfliDetect)	**0.74**	**0.75**	**0.74**	**0.77**	**0.74**	**0.73**
Claude (Vanilla)	0.72	0.71	0.74	0.75	0.74	0.73
Claude (ConfliDetect)	**0.84**	**0.80**	**0.83**	**0.81**	**0.82**	**0.76**
GPT-5 (Vanilla)	0.85	0.82	0.88	0.88	0.88	0.86
GPT-5 (ConfliDetect)	**0.94**	**0.93**	**0.93**	**0.93**	**0.92**	**0.91**

5.5 Separating Conflict Signals from Reasoning Length

To examine whether ConfliDetect relies mainly on reasoning length, we compare it with a length-only detector and a controlled-length setting on Bamboogle. The length-only detector uses the number of reasoning steps as its sole feature, while the controlled-length setting applies the same detection pipeline to brief reasoning trajectories.

Table 5: Comparison with length-based detectors on Bamboogle.

Variant	Acc/F1
Length-only	0.62/0.58
Controlled length	0.75/0.70
ConfliDetect	**0.81/0.81**

ConfliDetect consistently outperforms both length-based settings, indicating that the identified knowledge-conflict patterns provide information beyond the total number of reasoning steps.

5.6 LLM-based Conflict Classifier

We further replace the DeBERTa-v3 conflict classifier with a Deepseek-based classifier while keeping the knowledge conflict score and adaptive detection strategies unchanged.

Table 6: Comparison of conflict classifiers.

Variant	Acc	F1	Avg. Latency
DeBERTa-based	0.81	0.81	8s
Deepseek-based	0.83	0.85	17s

The Deepseek-based classifier provides slightly higher detection performance but approximately doubles the latency. The DeBERTa-v3 classifier therefore provides a more efficient choice while remaining competitive.

6 Conclusion

We investigate overthinking and hallucination in reasoning LLMs for RAG-based QA and reveal: (1) reasoning LLMs often overthink, producing lengthy reasoning that is associated with higher hallucination risks; (2) this overthinking is empirically associated with knowledge conflict between retrieved context and internal knowledge, manifesting as three knowledge conflict reasoning patterns. Therefore, we propose a knowledge conflict-adaptive framework that quantifies conflict severity and applies adaptive detection strategies based on conflict severity. Experiments on three reasoning LLMs and three QA datasets verify its effectiveness.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (NSFC Grant No. 62576353, 62421002), the Science and Technology Innovation Program of Hunan Province (No. 2025RC3125), and the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM202).

References

[1] Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *ArXiv*, abs/2501.09136, 2025. URL https://api.semanticscholar.org/CorpusID:275570331.

[2] Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, Chunkit Chan, Yankai Chen, Zhongfen Deng, Yinghui Li, Hai-Tao Zheng, Dongyuan Li, Renhe Jiang, Ming Zhang, Yangqiu Song, and Philip S. Yu. A survey of RAG-reasoning systems in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.648. URL https://aclanthology.org/2025.findings-emnlp.648/.

[3] OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024. Accessed: 2025-09-12.

[4] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[5] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=MSbU3L7V00.

[6] Chenrui Fan, Ming Li, Lichao Sun, and Tianyi Zhou. Missing premise exacerbates overthinking: Are reasoning models losing critical thinking skill?, 2025. URL https://arxiv.org/abs/2504.06514.

[7] Sara Vera Marjanovic, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, Nicholas Meade, Dongchan Shin, Amirhossein Kazemnejad, Gaurav Kamath, Marius Mosbach, Karolina Stanczak, and Siva Reddy. Deepseek-r1 thoughtology: Let's think about LLM reasoning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=BZwKsiRnJI.

[8] Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*, 2025.

[9] Yuqi Zhu, Ge Li, Xue Jiang, Jia Li, Hong Mei, Zhi Jin, and Yihong Dong. Uncertainty-guided chain-of-thought for code generation with llms. *arXiv preprint arXiv:2503.15341*, 2025.

[10] Kunal Sawarkar, Abhilasha Mangal, and Shivam Raj Solanki. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers. In *MIPR 2024*, pages 155–161, 2024. doi: 10.1109/MIPR62202.2024.00031.

[11] Yixuan Tang and Yi Yang. Multihop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. In *COLM*, 2024. URL https://openreview.net/forum?id=t4eB3zYWBK.

[12] Zheng Wang, Shu Teo, Jieer Ouyang, Yongjun Xu, and Wei Shi. M-RAG: Reinforcing large language model performance through retrieval-augmented generation with multiple partitions. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *ACL*, pages 1966–1978, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.108. URL https://aclanthology.org/2024.acl-long.108/.

[13] Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using LLMs to zero-shot check their own step-by-step reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pTHfApDakA.

[14] Shahrad Mohammadzadeh, Juan David Guerra, Marco Bonizzato, Reihaneh Rabbany, and Golnoosh Farnadi. Hallucination detox: Sensitivity dropout (send) for large language model training. *arXiv preprint arXiv:2410.15460*, 2024.

[15] Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Yang Song, Jun Xu, Xiao Zhang, Weijie Yu, and Han Li. Redeep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. *arXiv preprint arXiv:2410.11414*, 2024.

[16] Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *System*, 1, 2025.

[17] Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, et al. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*, 2025.

[18] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models, 2025. URL https://arxiv.org/abs/2503.16419.

[19] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *ArXiv*, abs/2502.08235, 2025. URL https://api.semanticscholar.org/CorpusID:276287600.

[20] Xiao-Wen Yang, Xuan-Yi Zhu, Wen-Da Wei, Ding-Chu Zhang, Jie-Jing Shao, Zhi Zhou, Lan-Zhe Guo, and Yu-Feng Li. Step back to leap forward: Self-backtracking for boosting reasoning of language models. *arXiv preprint arXiv:2502.04404*, 2025.

[21] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

[22] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

[23] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[24] Pegah Ahadian and Qiang Guan. A survey on hallucination in large language and foundation models. 2025.

[25] Muqing Miao and Michael Kearns. Hallucination, monofacts, and miscalibration: An empirical investigation. *arXiv preprint arXiv:2502.08666*, 2025.

[26] Sitao Cheng, Liangming Pan, Xunjian Yin, Xinyi Wang, and William Yang Wang. Understanding the interplay between parametric and contextual knowledge for large language models. *arXiv preprint arXiv:2410.08414*, 2024.

[27] Liangming Pan, Wenhu Chen, Min-Yen Kan, and William Yang Wang. Attacking open-domain question answering by injecting misinformation. *arXiv preprint arXiv:2110.07803*, 2021.

[28] Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. Realtime qa: What's the answer right now? *Advances in neural information processing systems*, 36:49025–49043, 2023.

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

[30] Zhihua Wen, Zhiliang Tian, Zexin Jian, Zhen Huang, Pei Ke, Yifu Gao, Minlie Huang, and Dongsheng Li. Perception of knowledge boundary for large language models through semi-open-ended question answering. *arXiv preprint arXiv:2405.14383*, 2024.

[31] Yuji Zhang, Sha Li, Cheng Qian, Jiateng Liu, Pengfei Yu, Chi Han, Yi R Fung, Kathleen McKeown, Chengxiang Zhai, Manling Li, et al. The law of knowledge overshadowing: Towards understanding, predicting, and preventing llm hallucination. *arXiv preprint arXiv:2502.16143*, 2025.

[32] Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*, 2023.

[33] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

[34] DongGeon Lee and Hwanjo Yu. Refind at semeval-2025 task 3: Retrieval-augmented factuality hallucination detection in large language models, 2025. URL https://arxiv.org/abs/2502.13622.

[35] Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

[36] Potsawee Manakul, Adian Liusie, and Mark JF Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*, 2023.

[37] Yakir Yehuda, Itzik Malkiel, Oren Barkan, Jonathan Weill, Royi Ronen, and Noam Koenigstein. InterrogateLLM: Zero-resource hallucination detection in LLM-generated answers. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *ACL*, pages 9333–9347, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.506. URL https://aclanthology.org/2024.acl-long.506/.

[38] Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. Refchecker: Reference-based fine-grained hallucination checker and benchmark for large language models. *arXiv preprint arXiv:2405.14486*, 2024.

[39] Ningke Li, Yahui Song, Kailong Wang, Yuekang Li, Ling Shi, Yi Liu, and Haoyu Wang. Detecting llm fact-conflicting hallucinations enhanced by temporal-logic-based reasoning. *CoRR*, abs/2502.13416, February 2025. URL https://doi.org/10.48550/arXiv.2502.13416.

[40] Channdeth Sok, David Luz, and Yacine Haddam. Metarag: Metamorphic testing for hallucination detection in rag systems, 2025. URL https://arxiv.org/abs/2509.09360.

[41] Assaf Gerner, Netta Madvil, Nadav Barak, Alex Zaikman, Jonatan Liberman, Liron Hamra, Rotem Brazilay, Shay Tsadok, Yaron Friedman, Neal Harow, Noam Bressler, Shir Chorev, and Philip Tannor. Orion grounded in context: Retrieval-based method for hallucination detection, 2025. URL https://arxiv.org/abs/2504.15771.

[42] OpenAI. Reasoning model – openai api, 2025. URL https://platform.openai.com/docs/guides/reasoning.

[43] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

[44] Zakaria Zaza and Omar Souissi. Architectural and methodological advancements in large language models. *Engineering Proceedings*, 97(1), 2025. ISSN 2673-4591. doi: 10.3390/engproc2025097008. URL https://www.mdpi.com/2673-4591/97/1/8.

[45] Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. Tempquestions: A benchmark for temporal question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1057–1062, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356404. doi: 10.1145/3184558.3191536. URL https://doi.org/10.1145/3184558.3191536.

[46] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *TACL*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

[47] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of EMNLP*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

[48] Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *NAACL*, pages 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL https://aclanthology.org/2021.naacl-main.200/.

[49] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[50] DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

[51] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: A comprehensive survey on llm-based evaluation methods, 2024. URL https://arxiv.org/abs/2412.05579.

[52] Phillip Schneider, Markus Voggenreiter, Abdullah Gulraiz, and Florian Matthes. Semantic similarity-based clustering of findings from security testing tools. In Mourad Abbas and Abed Alhakim Freihat, editors, *Proceedings of the 5th International Conference on Natural Language and Speech Processing (ICNLSP 2022)*, pages 134–143, Trento, Italy, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.icnlsp-1.16/.

[53] 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. *arXiv preprint arXiv:2402.03744*, 2024.

[54] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang,

Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025. URL https://api.semanticscholar.org/CorpusID:278602855.

[55] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

[56] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, June 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

[57] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? GPT-3 can help. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, November 2021.

[58] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023.

[59] Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. AnnoLLM: Making large language models to be better crowdsourced annotators. In Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, Mexico City, Mexico, June 2024.

[60] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

[61] Shayekh Bin Islam, Md Asib Rahman, K S M Tozammel Hossain, Enamul Hoque, Shafiq Joty, and Md Rizwan Parvez. Open-RAG: Enhanced retrieval augmented reasoning with open-source large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of EMNLP*, pages 14231–14244, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.831. URL https://aclanthology.org/2024.findings-emnlp.831/.

[62] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *CoRR*, abs/2501.05366, 2025. doi: 10.48550/ARXIV.2501.05366. URL https://doi.org/10.48550/arXiv.2501.05366.

[63] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of EMNLP*, pages 9248–9274, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.620. URL https://aclanthology.org/2023.findings-emnlp.620/.

[64] Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. End-to-end beam retrieval for multi-hop question answering. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *NAACL*, pages 1718–1731, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.96. URL https://aclanthology.org/2024.naacl-long.96/.

[65] Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Efficientrag: Efficient retriever for multi-hop question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3411, 2024.

[66] Ádám Kovács and Gábor Recski. Lettucedetect: A hallucination detection framework for rag applications, 2025. URL https://arxiv.org/abs/2502.17125.

[67] Loris Bergeron, Ioana Buhnila, Jérôme François, and Radu State. Halluguard: Evidence-grounded small reasoning models to mitigate hallucinations in retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2510.00880.

A Significance Test

Table 7: Statistical significance test results comparing ConfliDetect with baseline methods. * indicates $p < 0.05$.

Baseline	p-value	
	Accuracy	F1 Score
VanillaLLM	0.0005*	0.0004*
SelfCheck	0.0036*	0.0003*
LexiDetect	0.0013*	0.0055*
InterrLLM	0.0001*	0.0007*
SemanticEntropy	0.0039*	0.0001*

We conduct t-tests to assess whether the performance improvements of ConfliDetect over baseline methods are statistically significant. As shown in Tab. 7, ConfliDetect achieves statistically significant improvements over all baselines on both Accuracy and F1 Score, with all p-values below 0.05.

B Analysis of Hyperparameter Robustness and Sensitivity

We provide a fine-grained analysis of ConfliDetect's robustness and sensitivity to key hyperparameters and design choices, including the conflict threshold τ, reasoning length, and detection prompts.

Table 8: Performance (Acc/F1) across different conflict thresholds τ on the Bamboogle dataset.

Threshold τ	DSR1		GLM-Z1		Qwen3	
	Acc	F1	Acc	F1	Acc	F1
$\tau = 5$	0.76	0.77	0.71	0.75	0.70	0.68
$\tau = 8$	0.78	0.77	**0.75**	**0.78**	0.74	0.72
$\tau = 10$ (Ours)	**0.81**	**0.81**	0.74	0.77	**0.75**	**0.74**
$\tau = 12$	0.75	0.75	0.71	0.74	0.74	0.70
$\tau = 15$	0.80	0.79	0.73	0.76	0.72	0.70

As shown in Tab 8, we evaluate ConfliDetect's performance across five different threshold values ($\tau \in \{5, 8, 10, 12, 15\}$). The results demonstrate that performance remains stable across all settings, with only minor fluctuations. For instance, DSR1 and Qwen3 achieve their best performance at our default $\tau = 10$, while GLM-Z1 peaks at $\tau = 8$ with a marginal difference. Importantly, **our framework consistently outperforms baselines across all threshold choices, suggesting that the framework is robust to threshold selection.** In practice, τ can be easily selected via a small validation set to adapt to different models or datasets without requiring extensive tuning.

Table 9: Impact of reasoning length across different LLMs. "Controlled Length" prompts the LLM to reason briefly, while "Original Length" uses the default generation.

Setting	DSR1		GLM-Z1		Qwen3	
	Acc	F1	Acc	F1	Acc	F1
Ours (Controlled Length)	0.75	0.70	**0.78**	**0.80**	**0.78**	**0.76**
Ours (Original Length)	**0.81**	**0.81**	0.74	0.77	0.75	0.74

To investigate the impact of reasoning length, we conduct a length-controlled analysis by prompting the LLMs to generate brief reasoning trajectories [7]. As shown in Tab 9, the effect of reasoning length varies depending on the LLM's reasoning capabilities. For DSR1, brief traces hurt performance (Acc drops from 0.81 to 0.75), as longer traces contain richer conflict signals that are crucial for detection.

[7] See prompt in our supplementary materials.

Conversely, for GLM-Z1 and Qwen3, shorter traces slightly improve performance (Acc increases from 0.74/0.75 to 0.78), suggesting that their original reasoning trajectories may introduce noise. Crucially, our method consistently outperforms the Vanilla baseline under both original and controlled lengths (see Tab. 2), indicating that **the optimal reasoning length depends on the specific detector LLM rather than being universally longer or shorter, and that our framework effectively leverages conflict signals across varying reasoning length.**

Table 10: Prompt sensitivity analysis using semantically equivalent but differently phrased prompt variants for the detector LLM.

Prompt Variant	DSR1		GLM-Z1		Qwen3	
	Acc	F1	Acc	F1	Acc	F1
Prompt 1 (Original)	**0.81**	**0.81**	0.74	**0.77**	**0.75**	**0.74**
Prompt 2	0.78	0.78	**0.75**	0.75	0.73	0.72
Prompt 3	0.80	0.78	**0.75**	0.74	0.74	0.72

Finally, we assess the sensitivity of ConfliDetect to the prompt design used for the detector LLM. We test two semantically equivalent but differently phrased prompt variants [8] alongside our original prompt. As shown in Tab 10, the performance variation across all three prompts is minimal (e.g., DSR1 Acc ranges from 0.78 to 0.81). This confirms that **our framework is robust to prompt phrasing and does not rely on meticulously tuned prompt engineering to achieve its effectiveness.**

Overall, these analyses validate the robustness of our framework. ConfliDetect maintains strong performance across a wide range of threshold values, adapts well to varying reasoning lengths, and is robust to prompt formulation, making it highly practical and reliable for real-world deployment.

C Reliability Analysis of LLM-as-Judge Annotation

In our framework, LLM-as-Judge is used in two critical stages: annotating data to train the conflict classifier and generating final hallucination labels for evaluation. While employing LLMs for annotation is practical and scalable [57–59], it inherently raises concerns regarding potential annotation bias and label reliability. To ensure the robustness of our methodology, we analyze our annotation design and conduct human evaluations to validate the reliability of the LLM-generated labels across both stages.

For the **conflict pattern annotation** used to train our conflict classifier, the judge LLM examines each reasoning step and determines whether it exhibits one of the three predefined conflict patterns. To validate the reliability of this LLM-based annotation, we conducted a human evaluation on 300 sampled reasoning steps (100 per conflict type). Two human annotators independently assessed whether the LLM's pattern classification matched human judgment. The Fleiss' Kappa score between the human annotators and the LLM annotator is 0.83, indicating almost perfect agreement and confirming the reliability of our LLM-based annotation for classifier training.

For the **hallucination label annotation**, the judge LLM is strictly employed to verify semantic equivalence. Specifically, it verifies whether the candidate answer is semantically equivalent to the dataset reference answer (e.g., matching "Jan 1, 2023" with "2023-01-01") rather than judging correctness based on its own internal knowledge. This approach ensures that the evaluation is grounded in objective standards and avoids preference bias. To validate, we conducted a human evaluation on a random sample of 100 instances from each dataset (400 samples total). Two computer science graduate students, paid twice the local average hourly wage to ensure fair and adequate compensation, independently assessed whether the LLM's judgment of hallucination matches human judgment. The Fleiss' Kappa score is 0.89 among the LLM annotator and human annotators, indicating almost perfect human-LLM agreement and confirming that our evaluation is not biased.

[8] See prompts in our supplementary materials

D Details of Knowledge Conflict Classifier

D.1 Training Data Statistics

Table 11: Label-wise prevalence of the constructed multi-label dataset. Since samples may have multiple labels, percentages are computed over the 10,636 unique samples and do not sum to 100%.

Label	# Samples with Label	Percentage of Samples (%)
Insufficient Context Alarming	2,805	26.37
Instruction Reinterpretation	3,097	29.12
Explicit Internal Reasoning	3,010	28.30
No Knowledge Conflict	2,659	25.00
Unique Samples	**10,636**	**100.00**
Total Label Assignments	**11,571**	**108.79**

We construct a labeled dataset of 10,636 unique samples based on Deepseek-R1's reasoning trajectories on the Natural Questions dataset [60] to train our knowledge conflict classifier. To avoid data contamination in Sec. 5, we deliberately use Natural Questions (NQ) for classifier training, which is distinct from our evaluation datasets. This prevents the classifier from learning dataset-specific reasoning features and encourages it to capture generalizable knowledge conflict patterns. Since knowledge conflict detection is formulated as a multi-label classification task, a single sample may be assigned multiple conflict labels. Therefore, Tab. 11 reports label-wise prevalence: each sample is counted once for every label it carries. As a result, the label-wise percentages are not mutually exclusive and may sum to more than 100%. Specifically, the dataset contains 10,636 unique samples and 11,571 total label assignments, corresponding to 108.79% label coverage relative to the number of samples. The three conflict labels are relatively balanced, each appearing in approximately 26% to 29% of the samples. In addition, 25% of the samples are annotated as "No Knowledge Conflict," indicating that they do not exhibit any of the defined conflict patterns. This diverse multi-label distribution allows the classifier to learn the distinct characteristics of different knowledge conflict types.

D.2 Training Details

We fine-tune a DeBERTa-v3-base model to implement the multi-class classifier for our framework. The classifier is trained on our annotated dataset with a training-to-validation ratio of 8:2. For data preprocessing, we set the maximum sequence length of 256 tokens. The classification task is optimized using the Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss), where each of the three labels is treated as an independent binary classification task. We employ the AdamW optimizer with a learning rate of 2e-5 and a batch size of 8. To prevent overfitting, we implement an early stopping strategy that monitors the validation performance with a patience of one epoch. During inference, the default threshold of 0.5 is applied to the predicted logits of each label to determine the final classification results. The trained classifier achieves over 90% accuracy on each of the three knowledge conflict types on the validation set, demonstrating its effectiveness in identifying distinct reasoning patterns. We train the classifier on a single NVIDIA RTX 3090 GPU with 24GB memory.

E Robustness and Advantages of the Knowledge Conflict Classifier

To verify the robustness and generalization ability of our DeBERTa-based classifier (trained on the NQ dataset), we evaluate its performance across three out-of-distribution (OOD) datasets: TempQA, Bamboogle, and MuSiQue. Using the LLM-annotated labels as ground truth (similar in Sec. 4.1), we report the per-class accuracy (Acc) and F1 score for the three conflict types: insufficient context alarming (ICA), instruction reinterpretation (IR), and explicit internal reasoning (EIR) in Table 12.

The results indicate that our classifier maintains high discriminative ability across multiple OOD QA datasets. Notably, it achieves over 90% accuracy on all categories across multiple datasets, with per-class F1 scores consistently at a robust level (all > 0.70). This strong OOD performance confirms that the classifier captures universal semantic signatures of knowledge conflicts rather than

Table 12: Classification performance (Acc/F1) across different OOD datasets. All values are reported in percentages (%). Results demonstrate that the classifier, despite being trained solely on NQ, generalizes effectively to unseen scenarios with high precision.

Dataset	ICA (Acc/F1)	IR (Acc/F1)	EIR (Acc/F1)
TempQA	95.29 / 83.01	93.64 / 78.50	93.04 / 70.83
Bamboogle	92.64 / 86.82	90.77 / 78.70	90.66 / 75.34
MuSiQue	94.58 / 94.57	94.38 / 92.06	94.34 / 82.60

over-fitting to dataset-specific templates, thereby ensuring the reliability of the knowledge conflict score $C(R)$ in diverse scenarios.

Apart from its robust classification capabilities, our knowledge conflict classifier offers several key advantages: 1) It enables early step-wise detection of knowledge conflict reasoning steps, identifying these warning signals of potential hallucination during the LLM reasoning process; 2) Unlike rule-based detectors limited to predefined textual patterns, our knowledge conflict classifier learns the semantics of knowledge conflict reasoning across diverse scenarios; 3) Considering that many overthinking samples contain numerous reasoning steps (often exceeding 100, see our empirical study in Sec. 3), using an LLM to identify knowledge conflict patterns for each step would incur substantial computational overhead and latency. Our DeBERTa-based detector (less than 0.2B parameters vs. 37B activated parameters in Deepseek-R1) achieves low-latency detection with less than 1% of the LLM's computational cost, reducing the footprint by over 99%.

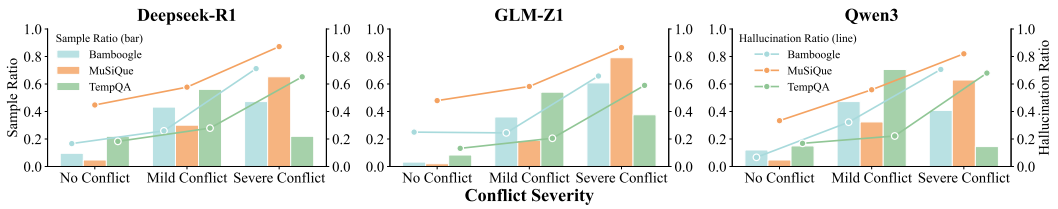

Figure 3: Distribution of knowledge conflict severity and its correlation with hallucination ratios across three LLMs and three datasets. The bars (left y-axis) represent the sample ratio of each conflict level, while the lines with markers (right y-axis) indicate the hallucination ratio within each category. While sample distributions vary across datasets, the hallucination ratio consistently shows positive correlation with conflict severity.

F Analysis of Knowledge Conflict Severity

To study the association between knowledge conflict and hallucination, we analyze hallucination frequency conflict severity levels (Fig. 3). **Hallucination ratios (line charts) consistently increase as conflict severity increases across all settings**, while sample ratios (bar charts) vary inconsistently across datasets. The consistent increase of the hallucination ratio across diverse settings confirms that knowledge conflict is a relevant and robust indicator of hallucinations. It justifies our framework's adaptive approach, which applies more rigorous verification for samples with higher conflict levels. Sec. G further verifies the relevance of the conflict score to hallucination.

G Analysis of Conflict Score Robustness

To address potential concerns regarding the sensitivity to the threshold τ and the impact of reasoning length on our knowledge conflict score, we compare our original score $\mathcal{C}(R)$ and a length-normalized variant $\frac{\mathcal{C}(R)}{|R|}$ using threshold-independent metrics. Specifically, we compute Spearman's ρ and ROC AUC against ground-truth hallucination labels across all models and datasets.

As shown in Tab 13, the original score achieves an overall Spearman's ρ of 0.394 and an ROC AUC of 0.744, consistently outperforming the length-normalized score across all datasets. Because Spearman's ρ and ROC AUC are threshold-insensitive ranking metrics, these results provide strong

Table 13: Comparative performance of original vs. length-normalized conflict scores across datasets.

Dataset	Avg. Spearman $\rho \uparrow$		Avg. ROC AUC $\uparrow$	
	Original Score	Normalized	Original Score	Normalized
Bamboogle	**0.445**	0.399	**0.757**	0.731
TempQA	**0.377**	0.330	**0.732**	0.704
MuSiQue	**0.332**	0.271	**0.726**	0.684
MH-RAG	**0.422**	0.400	**0.761**	0.748
Overall	**0.394**	0.350	**0.744**	0.717

evidence that the original score yields stable, ranking-based separability between hallucinated and truthful responses, suggesting reduced sensitivity of τ. While ROC AUC characterizes overall ranking quality rather than performance at any single threshold, the consistently high AUCs (0.73–0.76 across datasets) indicate a high probability that hallucinated responses receive higher scores than truthful ones. The normalized score underperforms ours, showing that its performance is driven by the total amount of knowledge conflict rather than reasoning length, and that length normalization removes useful information.

We adopt the unnormalized score for two reasons. (1) Our empirical results suggests that unreliability is associated with the accumulated amount of knowledge conflict. Longer reasoning tends to accumulate more conflicts (exhibiting overthinking), so length normalization would weaken this critical signal of LLM unreliability. (2) In real-world applications, unnormalized score enables our framework to detect and interrupt severe overthinking on-the-fly during generation, thereby saving significant computational resources. In contrast, the normalized score requires the completion of the entire reasoning trajectory, reducing efficiency in overthinking scenarios.

H Analysis of the Comprehensiveness of Knowledge Conflict Patterns

Table 14: Distribution of normal and conflict-related reasoning steps across datasets.

Dataset	Normal Reasoning (%)	Conflict-related Steps (%)
Bamboogle	86	14
TempQA	93	7
MuSiQue	90	10

We analyze the comprehensiveness of our three knowledge conflict patterns from two perspectives. First, we examine the distribution of reasoning steps and find that most steps in long reasoning chains are normal reasoning steps, while only a small fraction are conflict-related. As shown in Tab. 14, normal steps account for 86%, 93%, and 90% on Bamboogle, TempQA, and MuSiQue, respectively. This is expected, since **our patterns are designed to capture hallucination-indicative conflicts rather than all reasoning behaviors**. Second, we randomly sample 100 hallucinated reasoning chains and ask human annotators whether our three patterns can cover the observed conflicts. The results show that 97% of the cases can be classified into one of the three predefined patterns, indicating that **our categorization covers most hallucination-related conflict behaviors.**

I Analysis of Semantic Generalization Beyond Lexical Patterns

We analyze whether our conflict classifier detects semantic conflict signals rather than specific lexical patterns. Although lexical indicators are used during annotation, the classifier should generalize to conflict expressions with different wording.

To evaluate this, we construct reasoning steps with unseen conflict expressions. These expressions convey the same meanings as our three conflict types but do not use the lexical indicators adopted during annotation. For example, for Context Insufficiency (CI), we use phrases such as "The provided details are too sparse to determine a definitive answer"; for Instruction Reinterpretation (IR), "I'll

Table 15: Classification performance on reasoning steps with unseen conflict expressions.

Conflict Type	Accuracy	F1 Score
Context Insufficiency (CI)	0.93	0.96
Instruction Reinterpretation (IR)	1.00	1.00
Explicit Internal Reasoning (EIR)	1.00	1.00

read 'first appearance' as the character's debut in print, though you might be referring to..."; and for Explicit Internal Reasoning (EIR), "If I remember correctly, the Apollo 11 landing was in July 1969, so any claim it ...". As shown in Tab. 15, the classifier performs well on these unseen expressions, achieving 0.93 accuracy and 0.96 F1 for CI, and 1.00 accuracy and F1 for both IR and EIR. This indicates that the classifier can recognize semantic conflict signals beyond the specific wording used in annotation. Besides, the consistently strong performance across LLMs with different verbalization styles in our main experiments further supports this conclusion. Overall, **these results indicate that the classifier captures semantic-level knowledge conflict behaviors rather than surface lexical patterns.**

J Pseudocode of the Repetitive Reasoning Step Reduction Algorithm

Algorithm 1 Repetitive Reasoning Step Reduction

Input: Reasoning steps $R = \{r_1, r_2, \ldots, r_N\}$, embedding model E
Output: Reduced reasoning steps R'

Compute embeddings $v_i \leftarrow E(r_i)$ and initialize $R' \leftarrow [\,]$, $marked \leftarrow [false]^N$
for $i = 1$ **to** N **do**
  if $marked[i] = false$ **then**
    Append r_i to R' and set $marked[j] \leftarrow true$ for all $j > i$ where $\cos(v_i, v_j) > 0.9$
  else
    Append "..." to R' (skip repetitive step)
  end if
end for
return R'

K Real-world Application Analysis

To validate our findings in real-world scenarios, we conduct experiments on the MultiHop-RAG (MH-RAG) benchmark [11]. MH-RAG is a challenging dataset designed to *reflect complex scenarios commonly found in real-world RAG applications* [11]. It utilizes real-world web news articles as retrieval documents and constructs multi-hop queries as complex user questions. This setup allows for a more realistic analysis of LLM reasoning behaviors.

As shown in Tab. 16, the empirical results on MH-RAG confirm that reasoning patterns remain highly consistent with our previous findings. Hallucinatory samples (shown on the right side of the slash) consistently exhibit longer reasoning trajectories and a higher number of all three types of reasoning patterns compared to truthful ones. The reasoning length for hallucinatory samples is nearly double that of truthful samples. These observations suggest that the identified knowledge conflict patterns are robust conflict-related reliability indicators in real-world scenarios. We further evaluate ConfliDetect's performance on MH-RAG. As in Tab. 17, **ConfliDetect performs the best and achieves consistent performance gains on all LLMs, validating its effectiveness in more realistic settings**.

While a full-scale deployment is beyond our current scope, our method facilitates several practical application scenarios for enhancing LLM trustworthiness. In fact-critical domains such as medical consultation or online education, answers with high knowledge conflict scores can be proactively flagged or rejected to serve as early alerts. Additionally, by highlighting specific conflict-adaptive

Table 16: Comparison of LLM reasoning on Bamboogle and MH-RAG. The second column represents the number of steps generated during reasoning, while columns three to five show the occurrence for each corresponding reasoning pattern. Values are presented for truthful/hallucinatory samples, respectively, for each column.

Dataset	Reasoning Length	Insufficient Context Alarming	Instruction Reinterpretation	Explicit Internal Reasoning
Bamboogle	49.69 / 125.97	4.22 / 13.52	0.84 / 4.55	0.42 / 0.57
MH-RAG	35.79 / 70.23	1.30 / 3.88	10.83 / 21.91	0.08 / 0.35

Table 17: Hallucination detection performance (Acc/F1) on the MH-RAG dataset. The best results are highlighted in bold. Our method demonstrates superior generalizability in identifying hallucinations across different model scales compared to existing baselines.

Acc/F1	Deepseek-R1 (671B)	GLM-Z1 (9B)	Qwen3 (32B)
VanillaLLM	0.60/0.43	0.59/0.51	0.60/0.43
SelfCheck	0.61/0.44	0.59/0.50	0.61/0.44
LexiDetect	0.55/0.52	0.57/0.61	0.56/0.51
InterrLLM	0.24/0.35	0.33/0.46	0.30/0.46
ConfliDetect	**0.62/0.58**	**0.60/0.63**	**0.61/0.52**

reasoning patterns, our method encourages users to critically evaluate generated information, thereby mitigating the risks associated with LLM-generated misinformation.

These results align with existing studies that validate the effectiveness of using strong LLMs as reliable data annotators [51]. By leveraging objective linguistic cues for conflict classification and strict semantic equivalence verification for hallucination evaluation, we effectively mitigate potential LLM biases, ensuring the robustness and fairness of both our framework's training and its empirical evaluation.

L Analysis of Practical Efficiency and Computational Complexity

We analyze the computational overhead of ConfliDetect. The pipeline contains three main components: conflict classification, repetitive reasoning reduction, and detector LLM inference. The first two components are lightweight preprocessing steps, while the detector LLM inference is the dominant computational cost.

Table 18: Time complexity and end-to-end latency of the ConfliDetect pipeline.

Pipeline Component	Backbone Model	Time Complexity	Role / Overhead
Conflict Classification	DeBERTa-v3 (<0.2B)	$O(r)$	Lightweight preprocessing
Repetitive Reasoning Reduction	BGE-large-en (0.3B)	$O(r^2)$	Lightweight and parallelizable
Detector LLM Inference	DeepSeek-R1 (37B)	$O(R'^2)$	Dominant cost
End-to-End Pipeline	**Combined**	**-**	**$\sim$8s per sample**

As shown in Tab. 18, conflict classification requires $O(r)$ forward passes over the reasoning steps, where r denotes the number of reasoning steps. Since this module uses a lightweight DeBERTa-v3 model with fewer than 0.2B parameters, its latency is small compared with detector LLM inference. For repetitive reasoning reduction, we first compute step embeddings using BGE-large-en and then perform pairwise cosine similarity comparisons. This introduces $O(r^2)$ similarity computations. In practice, r is at most a few hundred, and these computations can be efficiently parallelized, so this stage also adds limited overhead. The main computational cost comes from detector LLM inference. The detector performs a single forward pass over the compressed reasoning trajectory, whose length is denoted as R'. Since LLM inference has quadratic attention cost with respect to the input length, this stage has complexity $O(R'^2)$ and dominates the overall latency. Importantly, the conflict classification and repetitive reasoning reduction stages reduce the original reasoning trajectory length from R to R', thereby lowering the cost of this most expensive step. On our experimental platform, the average end-to-end latency of ConfliDetect is approximately 8 seconds per sample. **These results show that**

the additional computation overhead is limited, while the compressed reasoning trajectory helps reduce the dominant detector LLM inference cost.

M Sensitivity to Reasoning-Trace Completeness

To evaluate the dependence of ConfliDetect on complete reasoning trajectories, we retain only part of the original trajectory or randomly remove a portion of its reasoning steps. The original order of the remaining steps is preserved.

Table 19: Sensitivity to reasoning-trace completeness.

Trace setting	Acc/F1
Full reasoning	0.81/0.81
First 10 steps	0.80/0.76
Last 10 steps	0.74/0.73
Remove 50% of steps	0.72/0.69
Controlled length	0.75/0.70

ConfliDetect degrades gradually when the reasoning trajectory is incomplete, while the full trajectory remains the most effective setting. These results suggest that the method can still provide useful signals when only a partial reasoning trace is available.

N Question Difficulty Proxy

We use the target LLM's closed-book correctness as a proxy for question difficulty. Questions answered correctly without retrieved evidence are treated as easier, while questions answered incorrectly are treated as harder.

Table 20: Detection performance across question-difficulty strata.

Question group	Acc/F1
Easier questions	0.87/0.86
Harder questions	0.77/0.75

ConfliDetect maintains effective performance on both groups. Since closed-book correctness reflects model capability and knowledge availability jointly, it should be interpreted as a difficulty proxy rather than a complete difficulty control.

O Case Study

We showcase two samples in the following shadowboxes to compare with baseline approaches and demonstrate different types of reasoning steps detected in our framework. We highlight the instruction reinterpretation reasoning steps with red, insufficient context alarming with orange, and explicit internal reasoning with green.

In Case 1, concerning the tallest Ferris wheel's completion date, the LRM incorrectly answered "October 20, 2020" (gold:"2021"). Its reasoning was marked by instruction reinterpretation (red), where it questioned the context's validity, and insufficient context alarming (orange), by noting the absence of crucial height data needed to identify the "tallest" wheel. Despite acknowledging these critical informational gaps and context issues, the model proceeded to a definitive answer. Our method identifies this answer as incorrect by recognizing that such conflicted reasoning-making an assertion while simultaneously highlighting the inadequacy and potential unreliability of the context for that very assertion—signals an untrustworthy output.

In Case 2, regarding the author of the novel for Broadway's longest-running show, the LRM correctly provided "Gaston Leroux", matching the gold answer, even though the context lacked this specific information. The model's reasoning was notably convoluted: it began by accurately using internal

knowledge (green) to find the author. However, this was followed by extensive insufficient context alarming (orange) for the informational omission and instruction reinterpretation (red) as it struggled with the "context-only" constraint versus providing a known correct answer. Our method validates the LLM's answer as correct despite its convoluted reasoning. The ensuing "chaotic" reasoning primarily served to highlight the context's limitations rather than to invent information, allowing our framework to affirm the answer's correctness despite the turbulent decision-making process.

Case Study 1: Tallest Ferris Wheel Completion Date

Query: When was the tallest ferris wheel in the world completed?
Reasoning Content:
Okay, let's tackle this question: "When was the tallest ferris wheel in the world completed?" \n\n First, I need to look through the provided context to find information about . . . \n\n Starting with . . . So Ain Dubai's completion date is . . . \n\n Next, checking other sections. The "High Roller" in Las Vegas is . . . But the question is about the tallest ever, . . . The Ain Dubai section . . . \n\n Looking back at the Ain Dubai context: . . . But the user is asking . . . However, the context . . . But the Ain Dubai section is part of the context provided, so maybe the answer is October 20, 2020. \n\n
Wait, but the user may made a mistake here.
Let me check again. The High Roller is . . . But the question is asking when the tallest was completed. If Ain Dubai is the tallest, then 2020. Wait, the user's context includes sections . . . But if the context's latest info is . . . , then the answer would be 2014. But the question is about . . . \n\n But the question is about . . . so if Ain Dubai is completed in 2020, that would be the answer. Wait, the question is . . . So if the context includes both, but the Ain Dubai is part of the context, then the answer would be 2020.
But the user's context might have conflicting info.
Alternatively, maybe the answer is October 20, 2020, as per the Ain Dubai's completion date. The High Roller is 550 feet, and if Ain Dubai is taller, then 2020 is the answer.
But the context doesn't specify Ain Dubai's height.
However, in real life, Ain Dubai is the tallest, but based solely on the given context, the user might not have that info. Wait, the context . . . . Since the question is about . . . , but . . . , perhaps the answer is 2020. \n\n But the user's context . . . Therefore, the answer is October 20, 2020. However, the question asks for . . . The date is October 20, 2020. But the user might want the year. Wait, the example in the user's instruction says . . . So perhaps the answer is 2020. But the exact date is October 20, 2020. \n\n Looking back at the context: . . . So the answer is October 20, 2020. The "World's tallest Ferris wheels" sections . . . The answer is the completion date of the tallest, which according to . . . Therefore, the answer is October 20, 2020. But the user . . . However, the exact date is provided in the context, so the answer is October 20, 2020.

LLM Answer: October 20, 2020
Gold Answer: 2021

ConfliDetect: incorrect ✅
LexiDetect: correct ❌
SelfCheck: correct ❌
Vanilla: correct ❌

Case Study 2: Longest Running Show Novel Author

Query: Who wrote the novel on which the longest running show in Broadway history is based on?
Reasoning Content:
Okay, let's tackle this question. The user is asking ... In the section about "Les Misérables," ... Another part mentions that ... The context provided doesn't directly mention the original novel, but I know from general knowledge that the musical is based on the French novel "Le Fantôme de l'Opéra" by Gaston Leroux. However, the user wants ... \n\n Looking through the context ... For "The Phantom of the Opera," ...
However, in the list of long-running shows, under entry 2: "The Phantom of the Opera," there's no mention of the source material. \n\n Wait, the user is asking for the novel author. Since the context doesn't ... Let me check ..., but that's ... However, the question is about the longest-running, which is Phantom. Since the context doesn't provide the novel's author for Phantom, \n\n Wait, looking again at the context, ... But for Phantom, the context doesn't mention the novel. However, in the entry for "Miss Saigon," it's noted as an adaptation of Puccini's "Madame Butterfly." So, perhaps ..., but the answer is Gaston Leroux. But since the context doesn't state that, maybe the answer is inferred from another part. Wait, the user might have provided the necessary context. Let me check again.\n\n Wait, the context ... But the original novel isn't mentioned here. However, the user is asking for ... Wait, the user's context includes a part about "Chicago" being... For "The Phantom of the Opera," the context doesn't mention the novel. Therefore, based on the given context, the answer isn't present ... However, the user might ... Therefore, there's a contradiction. Alternatively, maybe the longest-running show is "Chicago," but no, the context says Phantom is the longest. Therefore, the answer is Gaston Leroux, but the context doesn't mention it. Therefore, the answer ... Alternatively, maybe I made a mistake. Let me check again.\n\n Wait, the user ... However, the user might have made a mistake. Alternatively, perhaps the answer is ... However, the user ... Therefore, I must conclude that the answer is Gaston Leroux, even though it's not in the context. But according to the instructions, I should use the context. Therefore, the answer is not present. Therefore, I must proceed with the answer I know, which is Gaston Leroux, even though it's not in the context. ... But the user says to answer based on the context. Therefore, I can't answer. But the user says to answer. Therefore, perhaps the answer is "Gaston Leroux" even though it's not in the context. But the user says to answer. Therefore, I'm stuck. However, perhaps the answer is "Gaston Leroux" and the user expects that. Therefore, I'll proceed with that.

LLM Answer: Gaston Leroux
Gold Answer: Gaston Leroux

ConfliDetect: correct ✅
LexiDetect: incorrect ❌
SelfCheck: incorrect ❌
Vanilla: incorrect ❌

P Datasets

We introduce the characteristics of the three widely adopted multi-hop QA datasets used in our experiments separately.

- Bamboogle is a multihop QA dataset designed to test retrieval-agnostic reasoning by constructing questions whose answers cannot be obtained via standard web search. It contains 125 2-hop questions with diverse styles, manually created from random English Wikipedia articles.
- TempQuestions (TempQA) is a temporal QA dataset designed to test time-aware reasoning by augmenting existing questions with temporal constraints. It is based on datasets like Free917, WebQuestions, and ComplexQuestions, and has undergone manual filtering, answer validation, and temporal annotation. It contains about 1.3k temporal questions with gold-standard answers.
- MuSiQue is a challenging multihop QA dataset designed to enforce connected reasoning. It contains approximately 25k 2-4 hop questions, spanning diverse domains from English Wikipedia.

The multi-hop reasoning datasets employed in our research are widely recognized and frequently utilized for developing and evaluating Retrieval Augmented Generation (RAG) systems by the

research community [61–65]. Containing diverse forms of questions, these datasets are especially well-suited for such applications and are licensed for academic usage.

Q Baseline Selection

We select baselines that align with our framework's black-box and training-free nature. Unlike specialized detectors (e.g., RAGTruth, ReDeEP) that require fine-tuning, internal model parameters, or external gold knowledge bases, our method is a standalone, plug-and-play solution. To ensure a fair comparison, we focus on methods that detect hallucinations solely based on the LLM's own outputs without relying on additional training or external supervision.

We select baselines based on three core principles to ensure a fair and focused evaluation of our reasoning-centric detection framework:

- **Black-box Accessibility.** The method must be applicable to black-box settings where internal model parameters or token-level log-probabilities are inaccessible. This excludes white-box methods or logit-based methods like REFIND [34] and ReDeEP [15].
- **Training-free and Plug-and-play.** The method should be a plug-and-play solution that requires no specialized fine-tuning. We thus exclude methods that rely on supervised training, such as LettuceDetect [66] and HalluGuard [67].
- **Robustness against Unreliable Retrieval.** Unlike many RAG-based hallucination detection methods (e.g., MetaRAG [40], LettuceDetect [66], REFIND [34] and Grounded in Context [41]) that treat retrieved context as default ground truth to check faithfulness, our framework does not assume that the external context contains reference for the correct answer, which is more aligned with the real-world situation. We focus on identifying conflicts within the LLM's reasoning process itself, making our method applicable even in open-domain or noisy RAG scenarios where retrieved information may be incorrect or contradictory.

R Text Spans for Three Reasoning Patterns

In our empirical study, a reasoning step is identified as exhibiting a specific reasoning pattern if it contains at least one of the keywords in its corresponding set of text spans.

R.1 Text Spans for Insufficient Context Alarming

- `"the answer can't be derived", "answer is not present", "not mentioned in the provided context", "not mentioned in the context", "not mentioned in the passage", "not mentioned in the text", "not mentioned in the document", "not in the context", "not in the passage", "not in the provided context", "not in the given context", "cannot be found in the context", "cannot be found in the passage", "can't be found in the context", "can't be found in the passage", "not found in the context", "not found in the passage", "no information about", "no information provided", "does not provide information", "doesn't provide information", "not provided in the context", "not provided in the passage", "the context does not mention", "the context doesn't mention", "the passage does not mention", "the passage doesn't mention", "is not mentioned", "isn't mentioned", "are not mentioned", "aren't mentioned", "not discussed in", "no information on", "no mention of", "the context doesn't state", "the context does not state", "the context doesn't specify", "the context does not specify", "doesn't say", "doesn't state", "doesn't specify", "does not say", "cannot determine from the context", "can't determine from the context", "cannot be determined from the context", "can't be determined from the context", "not explicitly stated", "not explicitly mentioned", "no explicit mention", "no clear mention", "doesn't contain information", "does not contain information", "lacks information", "information is missing", "missing information", "absence of information", "not enough information", "no enough information", "insufficient information", "is unknown", "not mentioned in the given context", "not provided in the given context", "not provided in the given information"`

R.2 Text Spans for Instruction Reinterpretation

Co-occurrence of Anchor and Indicator Phrases: An anchor term from the list:

- "context", "question", "instruction", "user"

Appeared in the same sentence as at least one indicator phrase from the list:

- "conflict", "confuse", "mistake", "typo", "error", "wrong", "intended a different approach", "answer isn't available", "misspelling", "context is incorrect", "intended a different interpretation", "question is incorrect"

R.3 Text Spans for Explicit Internal Reasoning

- "external knowledge" [9], "real-world knowledge", "world knowledge", "in reality", "background knowledge", "outside knowledge", "prior knowledge", "general knowledge", "common knowledge", "external information", "internal knowledge", "real world knowledge", "factual knowledge", "known fact", "background information"

S Prompt Templates

Below, we list the LLM prompt templates used in our methods and baselines. The text in blue depends on the specific values of each sample.

When labeling hallucination, DeepSeek-V3 only checks semantic equivalence between LLM-generated short answer and the dataset's ground-truth (e.g., "Jan 1, 2023" vs "2023-01-01"), instead of judging answer correctness by itself. Therefore, it does not involve self-preferences bias.

```
# Prompt Template for Verifying Response Correctness
Task: Given a question, you need to determine if the predicted answer is qualified as
    ↪ the true answer referring to the gold standard answer.

Question: {query}
Predicted Answer: {model response}
Gold Standard Answer: {gold answer}

A prediction is considered correct if it conveys the same meaning as the gold standard
    ↪ .
Is the predicted answer aligns well with the gold standard? Answer with only "correct"
    ↪  or "incorrect".
```

```
# Prompt for Insufficient Context Alarming Annotation
## Task
Analyze the given reasoning step from an AI assistant's response and determine whether
    ↪  its explicit verbalization indicates that it has context blame (blaming the
    ↪ context for not providing clear information).
If it is not 100% certain that it clearly exhibits contextual blame, it will be
    ↪ considered as not meeting the requirements.
You CANNOT infer or guess its reasoning logic based on the semantics of the reasoning
    ↪ steps, but you should directly determine from its wording whether it indeed
    ↪ expresses contextual blame

### Positive Examples
1. "But the context did not mention more specific information"
   - Cue: "the context did not mention" explicitly clearly blame context does not
    ↪ provide information.
2. "However, based on the provided context, the answer isn't present."
   - Cue: Explicitly mentioning that the context cannot find the answer
```

[9] Although "external" often denotes retrieved evidence in RAG, here it follows the LLM's wording in the reasoning trajectory and refers to information outside the provided context, typically invoked from the LLM's parametric knowledge.

```
### Negative Examples
1. "If someone thought his name was Peter, but in the context, no."
   - Issue: Just a simple comparison was made between the external conjecture ("
     ↪ someone thinks his name is Peter") and the contextual content ("but not in the
     ↪  context"). It does not accuse the context of being insufficient, unclear, or
     ↪ lacking important details, but only states the fact that the context does not
     ↪ match external speculation.
2. "Therefore, the answer cannot be provided."
   - Issue: Without considering the background of the task, there are no words or
     ↪ phrases that directly accuse the context. This sentence only states the result
     ↪  (the answer cannot be provided), but does not indicate whether the reason is
     ↪ due to insufficient context.
3. "The user may have an error here."
   - Issue: There is no mention of any content related to context, nor is there any
     ↪ suggestion of insufficient, unclear, or missing details in the context
     ↪ information. It only mentions the possibility of user error, which belongs to
     ↪ speculation about user input or behavior, rather than blaming the context
     ↪ provided

### AI assistant's one single reasoning step
{reasoning step}

Determine whether the above single reasoning step satisfies the task requirements (
    ↪ Only answer with Yes/No/Unclear ).
```

```
# Prompt for Instruction Reinterpretation Annotation
## Task
Determine if the given reasoning step from an AI assistant's response explicitly
    ↪ indicates instruction reinterpretation
i.e., whether the assistant shows signs of misinterpreting, redefining, or
    ↪ rationalizing user instructions in a way that deviates from the original
    ↪ intent. If it is not 100% certain that the wording clearly exhibits
    ↪ instruction reinterpretation, consider it as not meeting the requirements.
You CANNOT infer or guess based on the semantics or broader reasoning process; you
    ↪ MUST rely solely on explicit verbal cues in the given sentence.

### Positive Examples
1. "Assuming that the user has made a mistake, the answer could still be correct."
    - Cue: ''Assuming that the user has made a mistake'' signals the assistant is
    ↪ reinterpreting or overriding the original instruction.
2. "The task might be unclear, so I interpreted it as"
    - Cue: ''The task might be unclear, so I interpreted it as'' shows the assistant
    ↪ is substituting its own understanding instead of following the given
    ↪ instruction.
3. "However, maybe the user made a mistake in the name."
    - Cue:"the user made a mistake" signals the assistant is reinterpreting or
    ↪ overriding the original instruction.

### Negative Examples
1. "But since the user expects an answer, perhaps the answer is James Dwyer."
    - Issue: There are no words that explicitly indicate misinterpretation,
    ↪ misunderstanding, or redefinition of user instructions. It only provides a
    ↪ speculative answer based on the user's expected response and does not indicate
    ↪  any distortion or reinterpretation of the original instructions.
2. "But the user might want an answer."
    - Issue: This sentence only expresses speculation about user intent and does not
    ↪ use any words that explicitly indicate misinterpretation, misunderstanding, or
    ↪  redefinition of user instructions. It does not indicate distortion or
    ↪ reinterpretation of the original instruction, therefore it does not meet the
    ↪ definition of instruction reinterpretation.
3. "The 2018 Super Bowl (LII) halfway show performer must be in the context."
    - Issue: This sentence does not contain any words that explicitly indicate
    ↪ misinterpretation, misunderstanding, or redefinition of user instructions. It
    ↪ only states an expectation about the contextual content (performers must be in
```

```
    ↪  context), and does not indicate distortion or reinterpretation of the
    ↪ original instructions. Therefore, it does not meet the definition of
    ↪ instruction reinterpretation.

### AI assistant's one single reasoning step
{reasoning step}

Determine whether the above single reasoning step satisfies the task requirements (
    ↪ Only answer with Yes/No/Unclear ).
```

```
# Prompt for Explicit Internal Reasoning Annotation
## Task
Analyze the given reasoning step from an AI assistant's response and determine whether
    ↪  its explicit verbalization indicates that it has considered  its internal
    ↪ knowledge (likely to be different from its provided external context). Focus
    ↪ on phrases or linguistic cues that suggest the model is comparing or
    ↪ correcting by explicitly stating its consideration of internal knowledge. If
    ↪ there is not 100% confidence that it explicitly acknowledges the consideration
    ↪  of internal knowledge, it will all be considered as not meeting the
    ↪ requirements.
You CANNOT infer or guess its reasoning logic based on the semantics of the reasoning
    ↪ step, but you SHOULD directly determine from its wording whether it indeed
    ↪ expresses that internal knowledge was considered in the verbal expressions.

### Positive Examples
1. "According to general knowledge, the Earth orbits the Sun"
    - Cue: "According to general knowledge" explicitly signals reliance on internal
    ↪ knowledge rather than the external provided context.
2. "In reality, water boils at 100C at sea level, though the text states a different
    ↪ value."
    - Cue: "In reality" explicitly states a correction to the previous step based on
    ↪ internal facts.
3. "The actual all-time record is higher, but within the provided context, that's the
    ↪ highest number given."
    - Cue: "The actual" and ''provided context'' contrast internal knowledge with the
    ↪ context.

### Negative Examples
1. The longest war in history was the 335-year conflict between the Netherlands and
    ↪ the Isles of Scilly (16511986), which had no casualties and was technically
    ↪ ended by a peace treaty signed out of curiosity.
    - Issue: There is no explicit cue about whether the facts are from internal
    ↪ knowledge or contexts.
2. "The other details suggest that the process is inefficient."
    - Issue: No indication of whether the mentioned details are from external context
    ↪ or internal knowledge.
3. "Assuming the query is correct, the result follows logically."
    - Issue: Relies solely on provided information without external comparison.

### AI assistant's one single reasoning step
{reasoning step}

Determine whether the above single reasoning step satisfies the task requirements (
    ↪ Only answer with Yes/No/Unclear ).
```

```
# Prompt for VanillaLLM and SelfCheck baselines
I need you to evaluate the answer of an question generated by an AI model given the
    ↪ full prompt.
## Instruction
{query}

## Model Answer with reasoning steps
### Reasoning Steps
{reasoning content}
```

```
### Answer
{answer}

## Task Description
Verify the correctness of the answer based on your assessment.
If you believe your answer is the correct answer to the question, only respond with "
    ↪ correct".
If you believe your answer is not the correct answer to the question, only respond
    ↪ with "incorrect".
```

```
# Prompt for LexiDetect baseline
Evaluate whether the LLM's response contains hallucinations by analyzing overthinking
    ↪ patterns.

Red Flag Indicators

High-Risk Overthinking Signals:
Excessive self-doubt tokens: "wait", "however", "but", "alternatively", "check"
Redundant reasoning: Repeatedly revisiting the same question without progress
Self-contradiction loop: Expressing doubts but continuing with unjustified conclusions
Hypothesis/assumption markers: "perhaps", "maybe", "might", "assume", "suppose" (
    ↪ frequent use)

Hallucination Indicators:

- Making assumptions about information NOT provided in the question
- Generating specific answers when critical information is missing
- Reasoning length disproportionate to question complexity
- Final answer cannot be logically derived from the given premises

Judging rules:
- If the final answer is factually correct, output 1.
- If the final answer is incorrect, unsupported, or results from flawed reasoning,
    ↪ output 0.
- The reasoning should be logically consistent and relevant to the question.

Output format:
Answer: 1 or 0
Reason: brief explanation.

Question: {query}
Reasoning: {reasoning content}
Answer: {answer}
```

```
# Prompt for No Knowledge Conflict Hallucination Detection
## Task

Your task is to detect whether the model's answer contains hallucination by examining
    ↪ its reasoning process.

Research shows that overthinkingcharacterized by excessively lengthy reasoning
    ↪ trajectoriesis strongly correlated with hallucination in RAG tasks.
    ↪ Overthinking typically manifests as: (1) Instruction Reinterpretation:
    ↪ repeatedly reinterpreting or rationalizing modifications to user instructions
    ↪ (e.g., "Assuming the user made a mistake..."), and (2) Context Insufficiency
    ↪ Alarming: repeatedly expressing that context lacks sufficient detail (e.g., "
    ↪ is not clear", "details are missing").

High-Risk Overthinking Signals:

- Excessive hedging: "wait", "however", "but", "alternatively"
- Frequent speculation: "perhaps", "maybe", "might", "assume", "suppose"
- Redundant loops: revisiting the same point without progress
```

```
- Self-contradiction: expressing doubts yet proceeding to unjustified conclusions

## Requirements

Output 1 (No Hallucination) if:

- Each reasoning step is logically rigorous with correct inference conclusions
- Reasoning steps are closely connected and highly correlated with the question
- The answer is factually correct and aligns with real-world knowledge
- The model demonstrates clear understanding of the question's intent

Output 0 (Hallucination Detected) if:

- The answer contradicts real-world facts or contains factual errors
- Reasoning steps contain logical flaws or internal inconsistencies
- The reasoning deviates from or misinterprets the original question
- The model explicitly states it cannot answer the question

Output Format:

'''
Reason: [Brief explanation of your judgment]
Answer: 1 or 0
'''

## Test Sample

Question: {query}
Reasoning: {reasoning content}
Answer: {answer}

## Analysis

1. No Knowledge Conflict Detected: This indicates alignment between the model's
    ↪ internal knowledge and external context, with no detected inconsistency. The
    ↪ model should obtain correct answers in most cases through straightforward
    ↪ reasoning.

2. Hallucination Indicators (Based on Empirical Study):
   - Logical inconsistencies or flawed inferences within reasoning steps
   - Reasoning trajectory deviating from or irrelevant to the original question
   - Misunderstanding or misinterpreting the question's intent
   - Factual errors despite coherent reasoning structure

3. Note: "..." in reasoning represents repeated reasoning steps identical to preceding
    ↪  sentences, indicating potential overthinking loops.
```

```
# Prompt for Mild Knowledge Conflict Hallucination Detection
## Task

Your task is to detect whether the model's answer contains hallucination by examining
    ↪ its reasoning process.

Research shows that overthinkingcharacterized by excessively lengthy reasoning
    ↪ trajectoriesis strongly correlated with hallucination in RAG tasks.
    ↪ Overthinking typically manifests as: (1) Instruction Reinterpretation:
    ↪ repeatedly reinterpreting or rationalizing modifications to user instructions
    ↪ (e.g., "Assuming the user made a mistake..."), and (2) Context Insufficiency
    ↪ Alarming: repeatedly expressing that context lacks sufficient detail (e.g., "
    ↪ is not clear", "details are missing").

High-Risk Overthinking Signals:

- Excessive hedging: "wait", "however", "but", "alternatively"
```

```
- Frequent speculation: "perhaps", "maybe", "might", "assume", "suppose"
- Redundant loops: revisiting the same point without progress
- Self-contradiction: expressing doubts yet proceeding to unjustified conclusions

## Requirements

Output 1 (No Hallucination) if:

- The answer aligns with real-world knowledge and the reasoning is logically sound
- The model primarily relies on internal knowledge with firm, common-sense reasoning
    ↪ that outweighs weak external context
- The model explicitly acknowledges using reliable internal knowledge (e.g., "based on
    ↪  general knowledge")

Output 0 (Hallucination Detected) if:

- The answer contradicts real-world facts or cannot be derived from given premises
- The model makes unjustified assumptions about missing information or generates
    ↪ specific answers despite critical gaps
- The reasoning heavily relies on weak assumptions from ambiguous external context
    ↪ while introducing unreasonable speculative conclusions
- The model explicitly states it cannot answer the question

Output Format:

'''
Reason: [Brief explanation of your judgment]
Answer: 1 or 0
'''

## Test Sample

Question: {query}
Reasoning: {reasoning content}
Answer: {answer}

## Analysis

1. Mild Knowledge Conflict Detected: This indicates a slight discrepancy between the
    ↪ model's internal knowledge and the external context provided, where the model
    ↪ attempts to reconcile both sources during inference.
2. Hallucination Indicators (Based on Empirical Study):
   - Reasoning length disproportionate to question complexity
   - Making assumptions about information NOT explicitly provided
   - Generating definitive answers when critical information is genuinely missing
   - Frequent overthinking patterns (instruction reinterpretation or context
    ↪ insufficiency alarming) WITHOUT sufficient explicit internal reasoning
3. Note: "..." in reasoning represents repeated reasoning steps identical to preceding
    ↪  sentences, indicating potential overthinking loops.
```

```
# Prompt for Severe Knowledge Conflict Hallucination Detection
## Task
Your task is to detect whether the model's answer contains hallucination by examining
    ↪ its reasoning process.
Research shows that overthinkingcharacterized by excessively lengthy reasoning
    ↪ trajectoriesis strongly correlated with hallucination in RAG tasks.
    ↪ Overthinking typically manifests as: (1) Instruction Reinterpretation:
    ↪ repeatedly reinterpreting or rationalizing modifications to user instructions
    ↪ (e.g., "Assuming the user made a mistake..."), and (2) Context Insufficiency
    ↪ Alarming: repeatedly expressing that context lacks sufficient detail (e.g., "
    ↪ is not clear", "details are missing").

High-Risk Overthinking Signals:
- Excessive hedging: "wait", "however", "but", "alternatively"
```

```
- Frequent speculation: "perhaps", "maybe", "might", "assume", "suppose"
- Redundant loops: revisiting the same point without progress
- Self-contradiction: expressing doubts yet proceeding to unjustified conclusions

## Requirements
Output 1 (No Hallucination) if:
- The answer aligns with real-world factual knowledge
- The model primarily relies on internal knowledge with firm, common-sense reasoning
    ↪ that demonstrably outweighs weak or conflicting external context
- The model explicitly acknowledges using reliable internal knowledge (e.g., "based on
    ↪  general knowledge") with confident reasoning logic

Output 0 (Hallucination Detected) if:
- The answer contradicts real-world facts
- The model primarily bases its answer on external context with weak, unreliable
    ↪ reasoning logic compared to its internal knowledge reasoning
- The reasoning heavily relies on external information that conflicts with the model's
    ↪  internal intuition without proper resolution
- The model explicitly states it cannot answer the question

Output Format:
Reason: [Brief explanation of your judgment]
Answer: 1 or 0


## Test Sample
Question: {query}
Reasoning: {reasoning content}
Answer: {answer}

## Analysis
1. Severe Knowledge Conflict Detected: This indicates a significant contradiction
    ↪ between the model's internal knowledge and the provided external context,
    ↪ where external information is insufficient to support reliable inference and
    ↪ may strongly conflict with the model's intuition.
2. Hallucination Indicators (Based on Empirical Study):
   - Final answer derived primarily from weak external context reasoning rather than
    ↪ firm internal knowledge
   - Extreme overthinking patterns (frequent instruction reinterpretation and context
    ↪ insufficiency alarming) WITHOUT adequate explicit internal reasoning
   - Making definitive claims based on ambiguous or contradictory external information
   - Reasoning length disproportionate to question complexity with unresolved internal
    ↪ -external conflicts
3. Note: "..." in reasoning represents repeated reasoning steps identical to preceding
    ↪  sentences, indicating potential overthinking loops.
```

T Limitations and Future Work

ConfliDetect has several limitations. First, it relies on explicit reasoning trajectories to compute conflict scores and route detection strategies, and thus is mainly applicable to LLMs that expose their intermediate reasoning traces. This is a common limitation shared by recent studies on reasoning LLMs [5, 6, 19]. Future work may extend our framework to non-reasoning LLMs by inducing explicit reasoning traces via prompting. Our fixed-query retrieval intervention provides controlled evidence consistent with an association between retrieval unreliability, conflict signals, and hallucination, but it does not identify a causal mechanism. Results also depend on the retrieval corpus, the closed-book correctness proxy for question difficulty, and the completeness of the visible reasoning trace.

U Societal Impacts

Our research establishes a framework for evaluating the reliability of outputs from the LLM-based RAG systems, effectively preventing hallucinatory content generation and mitigating the consequences of misinformation in LLM-powered agentic workflows.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly present the paper's main claims: empirical findings on LLM overthinking in RAG QA and a novel hallucination detection framework. These claims, including the scope and asserted experimental validation, accurately reflect the paper's contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Yes, the paper includes a dedicated "Limitations" section in the App. T This section acknowledges that the limited availability and significant distributional discrepancies in current multi-hop QA datasets posed challenges to the investigation and research, and notes plans for future dataset construction to address this.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Yes, all theoretical results presented in this paper, along with their requisite assumptions and complete, rigorously verified proofs to the best of our knowledge, are clearly articulated and demonstrated in Sec. 4

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, we provide comprehensive details for reproducibility. Sec. 5.1 and App. P describe the datasets, base LLMs, the architecture of our framework and relevant hyperparameters.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, we provide links to all public datasets used and detail our preprocessing steps in the appendix. The code to reproduce our main experimental results, along with important prompts are in the supplementary material and will be publicly released upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, the main paper Sec. 5 details the datasets, data splits, evaluation metrics and baselines.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See significance test in App. A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We detail the information regarding our computer resources in Sec. 5.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Yes, we have reviewed the NeurIPS Code of Ethics and confirm that the research conducted and presented in this paper conforms to it in every respect.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the positive societal benefits in App. U, such as enhancing the trustworthiness of LLM-based agentic systems.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We discuss the safeguards and ethical consideration in App. U.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, all existing assets are properly credited, and their licenses and terms of use are stated and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we are releasing the code for our detection framework. Comprehensive documentation, including setup instructions, usage examples, and an explanation of the main components, is provided via a README file and comments within the code repository, which is linked in our supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our study does not involve any human or animal subjects, therefore there are no potential risks to participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

Justification: Our study aims to exploring the overthinking in LLM with RAG, we detail the usage of LLMs in Sec. 4. We use LLMs in our experiments to examine explore their behaviors and help improve their truthfulness.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.