From Detection to Diagnosis: Advancing Hallucination Analysis with Automated Data Synthesis

Yanyi Liu[1], Qingwen Yang[1], Tiezheng Guo[1], Feiyu Qu[2], Jun Liu[1], Yingyou Wen[1] *

[1]School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[2]Department of Computer Science, Dartmouth College, Hanover, NH 03755, USA
{2290175, 2290182, 2310725}@stu.neu.edu.cn, feiyu.qu.gr@dartmouth.edu, liujun@cse.neu.edu.cn, wenyingyou@mail.neu.edu.cn

Abstract

Hallucinations in Large Language Models (LLMs), defined as the generation of content inconsistent with facts or context, represent a core obstacle to their reliable deployment in critical domains. Current research primarily focuses on binary "detection" approaches that, while capable of identifying hallucinations, fail to provide interpretable and actionable feedback for model improvement, thus limiting practical utility. To address this limitation, a new research paradigm is proposed, shifting from "detection" to "diagnosis". The Hallucination Diagnosis Task is introduced, a task which requires models to not only detect hallucinations, but also perform error localization, causal explanation, and content correction. We develop the Hallucination Diagnosis Generator (HDG), an automated pipeline that systematically generates high-quality training samples with rich diagnostic metadata from raw corpora through multi-dimensional augmentation strategies including controlled fact fabrication and reasoning chain perturbation. Using HDG-generated data, we train HDM-4B-RL, a 4-billion-parameter hallucination diagnosis model, employing Group Relative Policy Optimization (GRPO) with a comprehensive reward function incorporating structural, accuracy, and localization signals. Experimental results demonstrate that our model surpasses previous state-of-the-art detection models on the HaluEval benchmark while achieving comparable performance to advanced general-purpose models. In comprehensive diagnosis tasks, HDM-4B-RL matches the capabilities of larger general models while maintaining a smaller size. This work validates the feasibility and value of hallucination diagnosis, providing an effective methodology for building more trustworthy and reliable generative AI systems.

Introduction

Large Language Models (LLMs) frequently generate content that appears plausible but is factually inconsistent with world knowledge or the provided source context, this phenomenon is known as "hallucination." (Ji et al. 2023)

This phenomenon critically undermines the reliable application of generative AI systems (Chen et al. 2024), particularly in high-risk domains like medical diagnosis (Kim et al. 2025) or legal judgment (Herrera-Tapias and Hernández Guzmán 2025) where outputs can inform critical decisions. Consequently, developing robust methods to identify and mitigate hallucinations is imperative for the trustworthy deployment of LLMs.

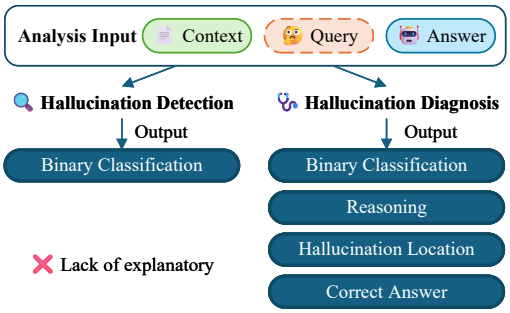

Figure 1: Contrasting hallucination detection with diagnosis.

This work focuses on "faithfulness hallucination," (Zhang et al. 2025b) a critical challenge where a model's output contradicts or lacks support from the provided source documents in tasks that demand strict fidelity, such as long-document summarization or knowledge-based QA.

A common approach is post-hoc detection (Liu et al. 2025). These methods often frame detection as a Natural Language Inference (NLI) task, assigning a coarse-grained label (e.g., Entailment, Contradiction) to the relationship between the response and its source.

However, we argue that the practical utility of this approach is limited. While a single classification label can flag an error, it fails to provide the fine-grained, interpretable feedback necessary for automated model correction or human-assisted revision. This limitation hinders the effective iterative refinement of generative models and fails to adequately address the core issue of user trust. Given the inherent challenges of eradicating hallucinations from within a model (Xu, Jain, and Kankanhalli 2024), we advocate for elevating the research paradigm from mere "detection" to comprehensive "diagnosis", as shown in Figure 1.

Analogous to medical diagnosis, an effective diagnostic system must not only determine whether a problem exists but also offer profound insights. We therefore define the "Hallucination Diagnosis Task," which requires a model to possess core capabilities across four dimensions:

*Corresponding author

- **Detection:** Accurately identifying if the generated content aligns with the source context.
- **Localization:** Pinpointing the specific segments within the output that are hallucinatory.
- **Explainability:** Explaining why the content is inconsistent.
- **Mitigation:** Proposing corrections or directly generating revised content that aligns with the source, based on the localization and explanation.

To facilitate this task, we began with the data, designing and implementing an automated pipeline for constructing a hallucination diagnosis dataset. Unlike methods that rely on existing NLI or QA datasets, our pipeline samples directly from large-scale pre-training corpora and applies multi-dimensional enhancement strategies—including controlled fact forgery, reasoning chain perturbation, and ambiguous information replacement, to automatically generate diverse diagnostic samples covering various task types and difficulty levels.

Finally, using the data produced by this pipeline, a 4B-scale hallucination diagnosis model, was trained and its effectiveness validated.

The main contributions of this work are as follows:

- A novel task, Hallucination Diagnosis Task, is proposed and formally defined with its core capabilities.
- The design and implementation of an automated pipeline to construct diagnostic datasets from large-scale pre-training corpora.
- A diagnosis model, HDM-4B-RL, is trained on the generated dataset, has proven effective across key diagnostic capabilities, thereby validating the proposed methods and data.

Related Work

Hallucination Detection

Existing methods for hallucination detection can be broadly categorized into two paradigms.

One line of research focuses on developing cost-efficient, specialized factuality classifiers. For example, methods like SummaC (Laban et al. 2022) adapt Natural Language Inference (NLI) models for document-level consistency checking. Other approaches, such as QAFactEval (Fabbri et al. 2022), leverage Question Answering (QA) frameworks to verify factual claims, while MiniCheck (Tang, Laban, and Durrett 2024) utilizes LLM-generated data for training its classifier, achieving state-of-the-art performance on real-world hallucination benchmarks.

The second paradigm leverages the inherent reasoning capabilities of Large Language Models (LLMs) for verification, often by decomposing the complex detection task into more manageable sub-steps. For instance, CoNLI (Lei et al. 2023) decomposes a claim and applies NLI to each segment. RaRR (Gao et al. 2022) verifies factuality by generating relevant questions coupled with external retrieval. Lynx (Ravi et al. 2024) utilizes the model's intrinsic reasoning capabilities to construct a factuality detection model through fine-tuning.

The present work moves beyond this binary fact-checking to introduce the more comprehensive task of "Hallucination Diagnosis." This task requires a model to not only detect an error but also to localize its position, provide natural language reasoning, and suggest a correction. We enable this integrated diagnostic capability through a novel, automated pipeline that synthesizes richly annotated diagnostic data from large-scale corpora.

Reasoning in Large Language Models

Significant progress has recently been made in eliciting and steering the reasoning capabilities of Large Language Models (LLMs). The seminal Chain-of-Thought (CoT) prompting method (Wei et al. 2022) demonstrated that prompting models to generate step-by-step thought processes dramatically improve their performance on complex tasks.

This insight has spurred a wealth of follow-up research. For example, Self-Consistency (Wang et al. 2022) enhances robustness by sampling diverse reasoning paths and taking a majority vote, while Tree-of-Thoughts (ToT) (Yao et al. 2023) generalizes this into a tree-like exploration that enables deeper planning and backtracking.

More recent advancements have pushed beyond prompt engineering to innovate on the models' intrinsic capabilities. Models such as DeepSeek-R1 (Guo et al. 2025), for instance, perform deep reasoning before generating a response, significantly enhancing their performance on complex tasks. Collectively, these works establish that modern LLMs possess a continuously evolving, intrinsic capacity for executing complex, multi-step logical operations.

This study builds directly on this progress, aiming to apply this powerful reasoning capability to the analysis of a model's own outputs, which is the core of the hallucination diagnosis task. The maturity and ongoing evolution of these reasoning capabilities are the key foundations that make this ambitious task technically feasible.

Methodology

This section details our proposed methodology for hallucination diagnosis. The approach comprises two core components: (1) the Hallucination Diagnosis Generator (HDG), a data synthesis pipeline, and (2) the Hallucination Diagnosis Model (HDM), a model trained on the data produced by HDG.

Hallucination Diagnosis Generator (HDG)

This section provides a detailed overview of our automated pipeline for constructing a hallucination diagnosis dataset, named the Hallucination Diagnosis Generator (HDG). The pipeline is designed to systematically generate a large-scale, high-quality, and richly annotated training dataset from raw pre-training corpora, specifically for training hallucination diagnosis models. Unlike datasets adapted from existing QA and NLI datasets, our pipeline generates more diverse samples with a broader difficulty distribution. As illustrated in Figure 2, the pipeline consists of four stages: task-oriented seed sample generation, multi-dimensional sample augmentation, quality verification, and metadata enrichment.

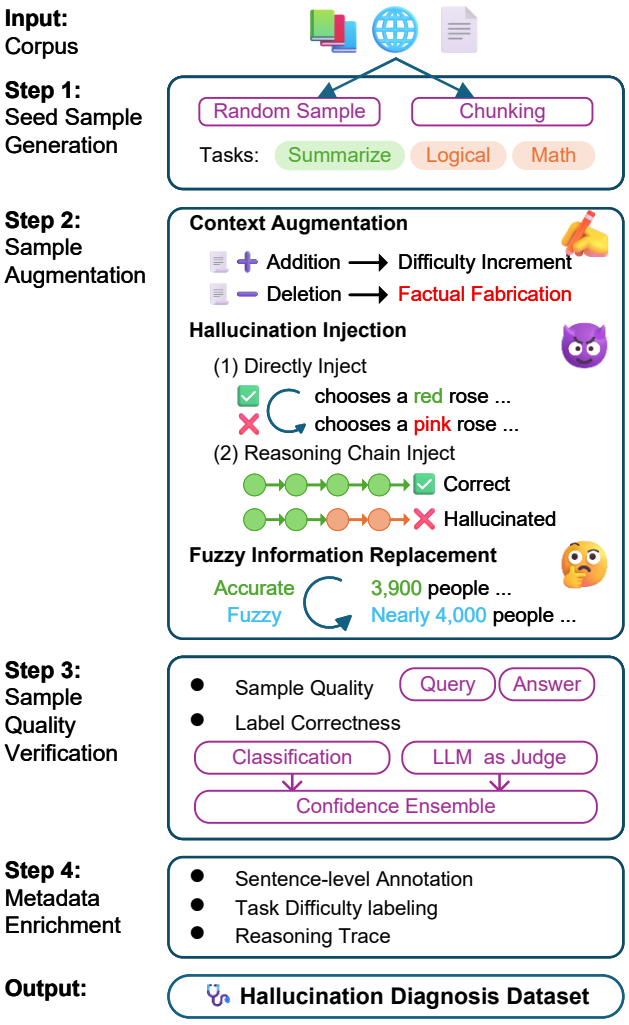

Figure 2: An overview of the Hallucination Diagnosis Generator (HDG) pipeline, detailing its four core stages from raw text to richly annotated diagnostic data.

Step 1: Task-Oriented Seed Sample Generation The objective of this stage is to generate high-quality "seed samples" from a large, unstructured corpus. Each sample, comprising a context, a query, and an answer, serves as the foundation for subsequent augmentation.

First, reference texts are sampled from a pre-training corpus and filtered using heuristic strategies (e.g., removing paragraphs that are too short or have high perplexity) to ensure informational richness and linguistic quality. For long documents, a recursive text splitter is employed to generate shorter context samples.

Diverse instructions are generated for the sampled texts, covering tasks such as summarization, logical reasoning, and mathematics to ensure a wide range of task types and difficulty levels.

Finally, responses for each instruction are generated using LLMs. For logical reasoning and mathematical tasks, we leverage the model's reasoning capabilities to maximize the accuracy of the generated answers. Specifically, we prompt the model to produce responses with a Chain-of-Thought (CoT) structure. The resulting seed samples are generated in two formats: one with a direct answer and another including the full reasoning chain.

Step 2: Sample Augmentation This stage represents the core innovation of our pipeline, designed to augment seed samples through a series of controllable perturbation strategies. Three primary augmentation strategies are employed: context augmentation, hallucination injection, and fuzzy information replacement.

Context Augmentation: Samples are enhanced by programmatically adding or deleting information in the context. Information addition introduces distractors to increase the difficulty of extracting relevant facts. Conversely, information deletion removes details pertinent to the original answer, thereby creating controlled scenarios of factual inconsistency.

Hallucination Injection: This strategy directly introduces errors into "correct" responses via two methods: 1) *Direct Injection*, applied to tasks like summarization, which creates factual errors by replacing key entities (e.g., "red rose" to "pink rose"); and 2) *Reasoning Chain Injection*, applied to logical and mathematical tasks, which generates logical hallucinations by perturbing a key step in a correct reasoning chain.

Fuzzy Information Replacement: This approach replaces precise information (e.g., 3,900 people) with vague expressions (e.g., nearly 4,000 people). This process creates outputs that are semantically plausible but not strictly faithful to the source. This type of augmentation is applied only to summarization tasks, as such ambiguity is unacceptable in reasoning and mathematical contexts.

Step 3: Sample Quality Verification This critical quality control stage ensures that augmented data pairs are linguistically fluent, logically coherent, and accurately labeled as either containing a hallucination or not.

Quality control involves two evaluation dimensions: **Task Sample Quality**, where an LLM scores the query-answer pair for clarity, fluency, and coherence (while ignoring correctness); **Hallucination Label Correctness**, where we use an ensemble-based approach to enhance labeling accuracy by integrating judgments from multiple models. An ensemble of top-tier classifiers (e.g., Lynx) and LLM-as-a-Judge methods (Gu et al. 2024) with a strong model (e.g., Qwen3-32B) integrate judgments based on confidence scores to ensure accuracy.

Step 4: Metadata Enrichment After quality verification, samples are enriched with structured metadata, transforming them from simple detection examples into comprehensive data suitable for training diagnostic models.

Core diagnostic labels include: **Sentence-Level Annotation**, which marks the specific location of hallucinations to train the model's localization capability; **Task Difficulty Labels**, which assigns a difficulty level based on the reasoning chain length required for the model to answer the original instruction; and **Reasoning Trace**, which provides an

explanatory chain for the final judgment.

The final output is a structured dataset where each entry contains the source context, instruction, response, hallucination location, difficulty, and a ground-truth answer, providing a robust foundation for training the HDM.

Hallucination Diagnosis Model

Fearing SFT could disrupt the base model's hybrid reasoning, we employed the Group Relative Policy Optimization (GRPO) (Shao et al. 2024) algorithm, using RL to guide the model's diagnostic capabilities while preserving its core abilities.

GRPO updates its policy by comparing the relative quality of a group of generated samples rather than relying on a single absolute score. This approach enhances training efficiency and stability while helping to preserve the model's native capabilities during the alignment process.

To translate the abstract task of "hallucination diagnosis" into a concrete, learnable objective for the model, a comprehensive, rule-based reward function is designed. This function assesses the model's output across multiple dimensions to guide it toward the desired diagnostic behaviors. The reward system consists of the following key components:

Structured Output Reward (R_{struct}) This foundational reward ensures a well-formed, machine-readable output, which is required to make other diagnostic components parsable. The model is incentivised to generate its diagnostic report as a JSON object containing four key fields: *conclusion*, *diagnosis*, *hallucinations*, and *corrected_answer*.

Each field is scored independently, and the final reward is the average of these scores. An incomplete or malformed JSON structure yields a reward of 0.

Detection Accuracy Reward (R_{acc}) This component treats hallucination detection as a binary classification task. It evaluates the accuracy of the model's judgment in the 'conclusion' field regarding whether the original answer contains hallucinations.

A positive reward is granted when the model's judgment aligns with the ground-truth label, calculated as follows:

$$R_{acc} = I(y = \hat{y}) \tag{1}$$

where $I(\cdot)$ is the indicator function (1 if true, 0 otherwise), y is the model's predicted conclusion (e.g., "pass" for no hallucination, "fail" for hallucination), and $\hat{y}$ is the corresponding ground-truth label.

Localization Reward (R_{loc}) This reward function evaluates the model's ability to locate and delineate hallucinatory content. The reward is calculated based on the degree of overlap between the set of predicted hallucinatory sentences and the ground-truth set.

Let S_{pred} be the set of sentences predicted as hallucinatory, and S_{gt} be the ground-truth set. A valid "hit" occurs when a predicted sentence $s_p \in S_{pred}$ has a string containment relationship with a ground-truth sentence $s_{gt} \in S_{gt}$ (i.e., s_p is a substring of s_{gt}, or vice versa).

In the event of a hit, a partial score is calculated based on the length ratio of the two sentences to penalize imprecise boundaries. Specifically, the scoring function $\text{score}(s_p, s_{gt})$ is defined as:

$$\text{score}(s_p, s_{gt}) = \begin{cases} \min\left(\frac{L_p}{L_{gt}}, \frac{L_{gt}}{L_p}\right) & \text{if hit}(s_p, s_{gt}) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where L_p and L_{gt} are the lengths of the predicted and ground-truth sentences, respectively. The score is 0 if there is no hit.

The final localization reward R_{loc} is defined as the sum of scores for each predicted sentence against its best-matching ground-truth sentence, normalized by the total number of ground-truth sentences to account for recall:

$$R_{loc} = \frac{1}{|S_{gt}|} \sum_{s_p \in S_{pred}} \max_{s_{gt} \in S_{gt}} \{\text{score}(s_p, s_{gt})\} \tag{3}$$

Final Reward Aggregation The final total reward R is the weighted sum of the above components, defined as:

$$R = \alpha_1 R_{struct} + \alpha_2 R_{acc} + \alpha_3 R_{loc} \tag{4}$$

The hyperparameters α_1, α_2 and α_3 are set to 1.0, 0.5, and 0.5, respectively, to balance the contribution of each component to the overall reward.

Dataset	Total	Halu	Non-Halu
QA Task (HaluBench)			
HaluEval (HE)	10000	5010	4990
RAGTruth (RT)	1000	160	740
FinanceBench (FB)	1000	500	500
DROP	1000	500	500
CovidQA (CQA)	1000	500	500
PubMedQA (PMQA)	1000	500	500
Summarization Task			
SummEval (SE)	1600	294	1306

Table 1: Distribution of samples in the evaluation datasets. The name in parentheses indicates the abbreviation used in subsequent sections.

Experiments

We designed a series of experiments to validate the effectiveness of our methodology across three core tasks: hallucination detection, localization, and mitigation.

Evaluation Datasets

Our evaluation framework covers two primary task categories: Question-Answering (QA) and Summarization. For QA tasks, we employ all subsets from the HaluBench benchmark (Ravi et al. 2024), which spans topics from general knowledge to specialized domains like finance and medicine. For the summarization task, we use the SummEval (Fabbri et al. 2020) for benchmarking. Detailed statistics for these datasets are presented in Table 1.

In addition to detection, we also evaluate the full scope of hallucination diagnosis, which includes localizing and mitigating hallucinatory content.

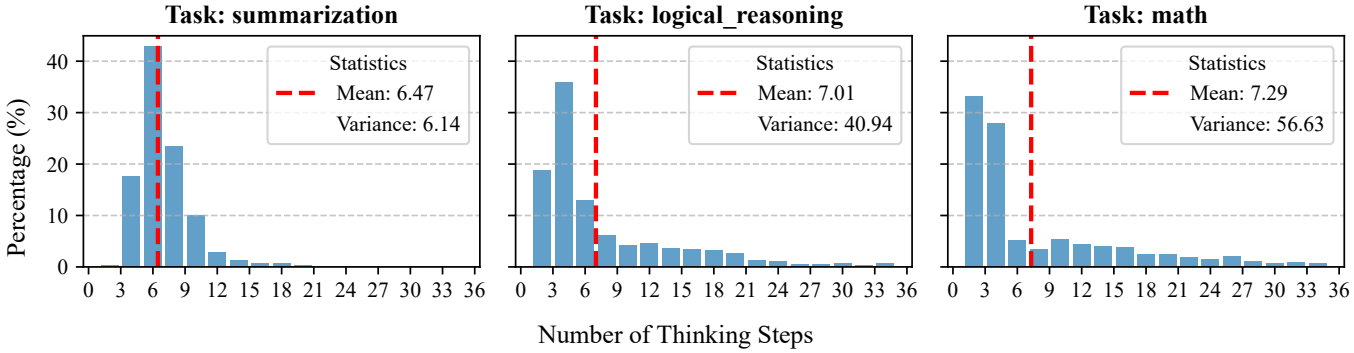

Figure 3: Distribution of reasoning steps in the synthesized hallucination diagnosis dataset, categorized by task type.

Task	Raw	Avg	Total
Summary	2332	5308	7640 (3236 / 4404)
Logical	988	6205	7193 (1613 / 5580)
Math	719	2901	3620 (1150 / 2470)
Total	**4039**	**14414**	**18453 (5999 / 12454)**

Table 2: Statistics of the constructed hallucination diagnosis dataset. The numbers in parentheses indicate the number of hallucination and non-hallucination samples, respectively.

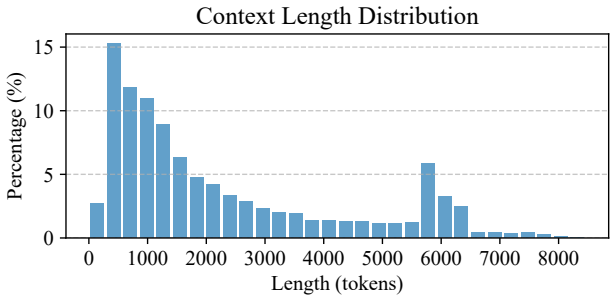

Figure 4: Distribution of context lengths in the constructed hallucination diagnosis dataset.

To this end, we constructed a dedicated diagnosis benchmark from the SummEval dataset. We first grouped all generated summaries by their reference text. From these groupings, we filtered for all summaries identified as containing hallucinations and further annotated these samples to pinpoint the specific sentences that were hallucinatory.

This process resulted in a diagnosis dataset of 285 samples, each containing a consistency label, sentence-level hallucination annotations, and ground-truth responses.

Experimental Setup

Data Construction We began by using our proprietary Hallucination Diagnosis Generator (HDG) pipeline to construct the training dataset. The data was sourced from the Wikipedia (20231101.en) dump, chosen for its breadth in establishing a general-purpose methodology, from which we randomly sampled 4,500 documents. Within the HDG pipeline, different models were employed at various stages based on a cost/efficiency/quality trade-off (e.g., efficient models for generation, strong ensembles for annotation).

The task generation stage utilizes Gemini-2.5-flash-lite, while the corresponding responses and injections are generated by Qwen3-32B (Yang et al. 2025). The context augmentation stage employs Qwen3-Embedding-0.6B (Zhang et al. 2025a) as an embedding model for similarity retrieval. In the quality verification stage, the judgment ensemble included Qwen3-32B (Yang et al. 2025), GPT-4.1, Lynx (Ravi et al. 2024), and MiniCheck-7B (Tang, Laban, and Durrett 2024). The subsequent metadata enrichment was performed by Qwen3-32B.

This process yielded a dataset of 18,453 samples, with the distribution across task types shown in Table 2. Analyses of the dataset's context length and difficulty distribution are presented in Figures 4 and 3, respectively.

The difficulty, measured by reasoning steps, increases from summarization to logical and mathematical reasoning. The latter two tasks also exhibit a larger variance, indicating a broader range of complexity.

Model Training Qwen3-4B is the base model for training HDM-4B-RL. The training was conducted using the ms-swift (Zhao et al. 2025) and Hugging Face Transformers (Wolf et al. 2020) libraries, employing the Group Relative Policy Optimization (GRPO) algorithm. We utilized 2 $\times$ NVIDIA A100 80GB GPUs. During training, one GPU was dedicated to the policy model for rollouts, while the other handled reward computations and policy model parameter updates. For training hyperparameters, The AdamW optimizer is used with a learning rate of 1×10^{-6} and a weight decay of 0.01. The he model was trained for a total of 2306 steps in batch-size 64.

Evaluation Setup For evaluation, open-source LLMs were deployed on a single NVIDIA A100 80GB GPU using the vLLM framework (Kwon et al. 2023). For models too large for a single GPU, such as Qwen3-32B, we applied FP8 quantization to enable deployment. Proprietary models were evaluated via their official API endpoints. Classifier-based models were loaded using the Transformers library and evaluated with their default inference configurations.

Model	Size	HE	RT	FB	DROP	CQA	PMQA	SE	Average
LLM Prompt Based									
Qwen3-4B (non-reasoning)	4B	73.64	58.25	56.72	52.29	78.26	68.83	66.43	64.92
Qwen3-4B	4B	75.15	61.28	82.98	76.19	87.61	77.72	76.26	76.74
Qwen3-32B (non-reasoning)	32B	80.92	57.83	69.59	70.61	83.94	83.43	72.61	74.13
Qwen3-32B	32B	77.63	69.36	86.58	**80.10**	92.99	87.10	**78.16**	81.70
GPT-4.1	-	82.95	62.58	60.79	66.91	90.75	**88.17**	75.41	75.37
o4-mini	-	**83.23**	**71.89**	**88.90**	76.28	**94.19**	85.95	77.88	**82.62**
Hallucination Detection Models									
HHEM	110M	66.59	**73.72**	40.85	46.55	57.48	57.22	61.00	57.63
Alignscore-Large	355M	74.10	56.77	41.93	50.90	64.11	61.05	65.60	59.21
FactCG-DeBERTa-v3-Large	435M	66.29	52.28	42.03	48.67	78.48	59.56	61.00	58.33
MiniCheck-Flan-T5-Large	783M	66.85	53.65	42.46	49.74	75.48	71.72	76.20	62.30
Bespoke-MiniCheck-7B	7B	79.73	53.08	50.38	64.78	88.10	64.75	**78.56**	68.48
Lynx-8B	8B	**86.96**	60.25	74.30	64.79	**97.70**	**88.37**	68.39	77.25
HDM-4B-RL (non-reasoning)	4B	82.84	63.74	57.83	50.85	84.21	80.67	70.75	70.13
HDM-4B-RL	4B	83.96	68.52	**84.54**	**74.39**	92.50	77.79	75.88	**79.65**

Table 3: Hallucination Detection Results. Macro F1 score is used for comparison. Bold and underlined values indicate the best and the second-best results in each group, respectively.

Hallucination Detection

To validate the capabilities of our model HDM-4B-RL, we conducted evaluations on the hallucination detection task. Following the HaluBench benchmark, we treat the hallucination detection task as a binary classification task, and use the macro F1 score as the primary metric.

For a comprehensive comparison, our evaluation includes two categories of baselines: (1) specialized hallucination detection models, including HHEM (Bao et al. 2024), Alignscore (Zha et al. 2023), FactCG (Lei et al. 2025), MiniCheck (Tang, Laban, and Durrett 2024), and Lynx (Ravi et al. 2024); and (2) prompt-based LLMs, for which we use Qwen3-4B, Qwen3-32B, GPT-4.1, and o4-mini. For models with a switchable reasoning mode, like Qwen3, we report results for both modes. The results are presented in Table 3.

Comparison with Hallucination Detection Models The results show that HDM-4B-RL is the top-performing specialized hallucination detection model, excelling on FinanceBench and DROP with F1 scores of 84.54 and 74.39, respectively. Compared to the previous SOTA model, Lynx-8B, our model achieved a 2.4% higher average F1 score despite being half the size. This demonstrates our method's superior balance between computational efficiency and performance, achieving better results with a smaller model.

Comparison with General LLMs We also acknowledge that a performance gap remains when compared to top-tier prompt-based models like o4-mini (82.62 F1). However, our approach shows immense potential. The performance of HDM-4B-RL (79.65 F1) already surpasses powerful closed-source models like GPT-4.1 (75.37 F1) and is closing the gap with the reasoning-enabled Qwen3-32B (81.70 F1). This indicates that through specialized fine-tuning, a smaller model can challenge unoptimized larger models on specific tasks, offering an efficient solution for resource-constrained scenarios. This is highlighted in detection latency, where the 32B model requires approximately $2.2\times$ the runtime of ours.

Reasoning Mode vs. Non-Reasoning Mode The performance benefit of the reasoning mode is a key finding of our experiments. For the Qwen3-4B model, enabling its reasoning mode boosted the average F1 score from 64.92 to 76.74. This effect was most pronounced on datasets like FinanceBench (heavy on calculation) and DROP (reliant on discrete reasoning), which align with the strengths of reasoning-focused models. This enhancement effect is equally significant on our fine-tuned model, HDM-4B-RL, with the F1 score rising from 70.13 (non-reasoning) to 79.65 (reasoning). Given that we performed RL fine-tuning only in reasoning mode, this indicates that our method does not compromise the model's performance in non-reasoning mode. Reasoning mode requires approximately $1.8\times$ the runtime of non-reasoning mode, presenting an accuracy-efficiency trade-off adaptable to specific needs.

Effectiveness of RL Training Table 4 provides direct evidence of the effectiveness of our fine-tuning. Compared to its base model, Qwen3-4B, HDM-4B-RL achieves comprehensive improvements. A deeper analysis reveals the source of this performance boost. In reasoning mode, the F1 score improvement (+2.91) is primarily driven by a significant gain in Recall (+4.62) and a solid increase in Precision (+2.50). This indicates that our model's ability to "identify all true hallucinations" has been greatly enhanced without sacrificing judgmental accuracy. More interestingly, in non-reasoning mode, the improvement in the F1 score is even larger (+5.21). This is driven almost entirely by a massive leap in Recall (+5.97), while Precision remains stable (-0.26). This suggests our fine-tuning strategy greatly en-

Model	P	R	F1	Acc
Lynx-8B	77.79	77.54	77.25	81.63
Qwen3-4B†	70.93	65.35	64.92	71.83
HDM-4B-RL†	70.67 (-0.26)	71.32 (+5.97)	70.13 (+5.21)	73.74 (+1.91)
Qwen3-4B	77.89	76.42	76.74	80.85
HDM-4B-RL	80.39 (+2.50)	81.04 (+4.62)	79.65 (+2.91)	82.24 (+1.39)

Table 4: Performance comparison with Qwen3-4B. P, R, F1, and Acc represent Precision, Recall, F1 score, and Accuracy, respectively. The † indicates evaluation in non-reasoning mode.

Model	Det	Loc		Mit
	Acc	HR	SV	AS
Original Result				59.77
Pipeline Method				
Qwen3-32B	94.74	76.97	69.69	79.55
GPT 4.1	61.75	60.52	53.08	76.77
o4-mini	82.46	78.59	79.71	78.51
Single Prompt Method				
Qwen3-32B	97.54	64.85	59.15	70.98
GPT 4.1	77.54	59.12	48.10	65.97
o4-mini	89.12	41.30	43.04	61.18
HDM-4B-RL	92.28	58.65	48.49	69.16

Table 5: Performance Comparison on the Hallucination Diagnosis Task. Det, Loc, and Mit represent detection, localization, and mitigation tasks. The "Original Result" serves as a baseline for the mitigation task.

hanced the model's fundamental pattern recognition capabilities, making it more sensitive to capturing the features of hallucinations even without relying on complex reasoning.

Hallucination Diagnosis Result

Lacking established public benchmarks for the full diagnosis task, the hallucination diagnosis task is conceptualized as a composite challenge, and its evaluation is conducted by deconstructing it into three core sub-tasks: detection, localization, and mitigation.

For the granular analysis, specific metrics are applied to each sub-task. For **detection**, given that this evaluation exclusively involves samples containing hallucinations, performance is measured directly by Accuracy (ACC). The **localization** capability is assessed by a "Hit Rate" (HR), analogous to the R_{loc} reward, and supplemented by a "Span Validity" (SV) metric to ensure localized spans are verbatim substrings of the original answer. Finally, the quality of the corrected text from the **mitigation** sub-task is evaluated using AlignScore (Zha et al. 2023) (AS), a reference-based metric for factual consistency.

To establish performance benchmarks, two baseline approaches based on prompt engineering were constructed. The first is the single-prompt method, which attempts to generate the entire diagnostic report in a single pass, our HDM-4B-RL is a representative of this method. The second is the pipeline method, which decomposes the task into a sequence of three separate steps: detection, localization, and mitigation. The final results are presented in Table 5

Table 5 reveals a distinct trade-off. While the multi-step pipeline achieves superior performance on fine-grained tasks like localization and mitigation via task decomposition, it incurs high latency due to multiple LLM calls. Conversely, the end-to-end method is far more efficient: using HDM e2e as the baseline ($1\times$ runtime), 32B e2e requires only $1.5\times$, whereas the 32B pipeline demands $4.9\times$.

Against this backdrop, the proposed HDM-4B-RL model is particularly noteworthy. Operating as a single-prompt model, it demonstrates a clear and comprehensive performance advantage over other baselines in its category, including GPT-4.1 and o4-mini. More importantly, as a lightweight 4B model, its performance is highly competitive across all diagnostic sub-tasks, closely approaching the quality achieved by the 32B-scale model, even in the most challenging mitigation task. This outcome strongly validates that a lightweight, end-to-end model, when subjected to specialized alignment with high-quality data, can achieve performance comparable to that of much larger counterparts that rely on more cumbersome processes. This provides a solid basis for developing AI systems that are both highly efficient and reliable.

Limitations and Conclusion

Limitations Despite promising results, our study has several limitations that highlight avenues for future research. First, regarding the data, our HDG pipeline currently relies solely on Wikipedia as its source corpus. Future work should expand these data sources to more specialized domains (e.g., scientific literature, legal documents) and further scale the dataset's size. Second, due to computational constraints, our experiments were confined to a 4B-parameter model. Exploring the effectiveness of our approach on larger-scale models is a key direction for future work.

Conclusion In this paper, we addressed the problem of hallucination in large language models by proposing a "hallucination diagnosis" paradigm that moves beyond traditional binary detection. We defined this task by four core capabilities—detection, localization, explanation, and mitigation. To tackle this, we introduced a novel methodology featuring the HDG automated data pipeline and trained an efficient Hallucination Diagnosis Model, HDM-4B-RL, using the GRPO algorithm on our synthesized data. Experiments demonstrated that this specialized 4B model not only outperforms larger, detection-specific models but also achieves highly competitive performance in the full diagnostic workflow, rivaling that of powerful general-purpose models. These results validate the efficacy of our approach, presenting a promising path toward developing more reliable and trustworthy AI systems.

Acknowledgments

This study is supported by Liaoning Provincial Science and Technology Innovation Project in the Field of Artificial Intelligence (Project name: Research on key technologies for systems engineering of large language model)(Grant no.2023JH26/10100005).

References

Bao, F.; Li, M.; Luo, R.; and Mendelevitch, O. 2024. HHEM-2.1-Open.

Chen, W.; Yan-yi, L.; Tie-zheng, G.; Da-peng, L.; Tao, H.; Zhi, L.; Qing-wen, Y.; Hui-han, W.; and Ying-you, W. 2024. Systems engineering issues for industry applications of large language model. *Applied Soft Computing*, 151: 111165.

Fabbri, A. R.; Kryściński, W.; McCann, B.; Xiong, C.; Socher, R.; and Radev, D. 2020. SummEval: Re-evaluating Summarization Evaluation. *arXiv preprint arXiv:2007.12626*.

Fabbri, A. R.; Wu, C.-S.; Liu, W.; and Xiong, C. 2022. QAFactEval: Improved QA-Based Factual Consistency Evaluation for Summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2587–2601.

Gao, L.; Dai, Z.; Pasupat, P.; Chen, A.; Chaganty, A. T.; Fan, Y.; Zhao, V. Y.; Lao, N.; Lee, H.; Juan, D.-C.; et al. 2022. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.

Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Herrera-Tapias, B. A.; and Hernández Guzmán, D. 2025. Legal Hallucinations and the Adoption of Artificial Intelligence in the Judiciary. *Procedia Computer Science*, 257: 1184–1189. The 16th International Conference on Ambient Systems, Networks and Technologies Networks (ANT)/ the 8th International Conference on Emerging Data and Industry 4.0 (EDI40).

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38.

Kim, Y.; Jeong, H.; Chen, S.; Li, S. S.; Lu, M.; Alhamoud, K.; Mun, J.; Grau, C.; Jung, M.; Gameiro, R.; et al. 2025. Medical hallucinations in foundation models and their impact on healthcare. *arXiv preprint arXiv:2503.05777*.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Laban, P.; Schnabel, T.; Bennett, P. N.; and Hearst, M. A. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10: 163–177.

Lei, D.; Li, Y.; Hu, M.; Wang, M.; and Yun, X. 2023. Chain of natural language inference for reducing large language model hallucinations. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Lei, D.; Li, Y.; Li, S.; Hu, M.; Xu, R.; Archer, K.; Wang, M.; Ching, E.; and Deng, A. 2025. FactCG: Enhancing Fact Checkers with Graph-Based Multi-Hop Data. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 5002–5020.

Liu, Y.; Yang, Q.; Tang, J.; Guo, T.; Wang, C.; Li, P.; Xu, S.; Gao, X.; Li, Z.; Liu, J.; et al. 2025. Reducing hallucinations of large language models via hierarchical semantic piece. *Complex & Intelligent Systems*, 11(5): 1–19.

Ravi, S. S.; Mielczarek, B.; Kannappan, A.; Kiela, D.; and Qian, R. 2024. Lynx: An open source hallucination evaluation model. *arXiv preprint arXiv:2407.08488*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Tang, L.; Laban, P.; and Durrett, G. 2024. MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 8818–8847.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Xu, Z.; Jain, S.; and Kankanhalli, M. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliber-

ate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Zha, Y.; Yang, Y.; Li, R.; and Hu, Z. 2023. AlignScore: Evaluating Factual Consistency with A Unified Alignment Function. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11328–11348. Toronto, Canada: Association for Computational Linguistics.

Zhang, Y.; Li, M.; Long, D.; Zhang, X.; Lin, H.; Yang, B.; Xie, P.; Yang, A.; Liu, D.; Lin, J.; et al. 2025a. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176*.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; Fu, T.; Huang, X.; Zhao, E.; Zhang, Y.; Chen, Y.; Wang, L.; Luu, A. T.; Bi, W.; Shi, F.; and Shi, S. 2025b. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *Computational Linguistics*, 1–45.

Zhao, Y.; Huang, J.; Hu, J.; Wang, X.; Mao, Y.; Zhang, D.; Jiang, Z.; Wu, Z.; Ai, B.; Wang, A.; Zhou, W.; and Chen, Y. 2025. SWIFT: A Scalable Lightweight Infrastructure for Fine-Tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28): 29733–29735.