

RADAM: A Retrieval-Augmented Detection and Mitigation Framework for Faithfulness Hallucinations

Bin He¹, Guanghua Li¹, Yu Liu², Mingyang Zhou¹,
Rui Mao^{1,3}, Naipeng Chao¹, Hao Liao^{1,4,*}

¹College of Computer Science and Software Engineering, Shenzhen University, China

²Department of Systems Science, Beijing Normal University, China

³Shenzhen Institute of Computing Sciences

⁴Provincial Key Laboratory of Intelligent Communication
and Digital Society Governance, Shenzhen University, China
haoliao@szu.edu.cn

Abstract

Retrieval-augmented generation (RAG) empowers large language models (LLMs) with external knowledge but remains vulnerable to faithfulness hallucinations—answers that contradict retrieved evidence. We introduce RADAM, a lightweight post-processing framework that mitigates this issue via fine-grained hallucination detection and guided revision. RADAM leverages a single diagnostic critic, optimized via preference tuning on taxonomy-guided data, to generate structured critiques encompassing both precise error spans and explicit revision suggestions, which guide the base model toward more faithful outputs. Experiments on real-world datasets demonstrate that RADAM consistently achieves the lowest average hallucination rate across the evaluated methods, while largely preserving response quality and achieving out-of-domain performance comparable to enterprise-grade LLMs.

1 Introduction

Recent studies have shown that large language models (LLMs) are unable to directly access up-to-date knowledge (Xu et al., 2025). To address this limitation, retrieval-augmented generation (RAG) has emerged as a popular paradigm that supplements LLMs with external knowledge bases (Fan et al., 2024). However, RAG systems introduce new challenges, among which faithfulness hallucinations has gained increasing attention (Huang et al., 2025; Ji et al., 2023). Faithfulness hallucinations refers to cases where the model’s output contradicts the retrieved context, undermining the reliability of the generated response (Ladhak et al., 2022; Madsen et al., 2024). The importance of mitigating such hallucinations has been widely recognized in existing research (Chuang et al., 2024b).

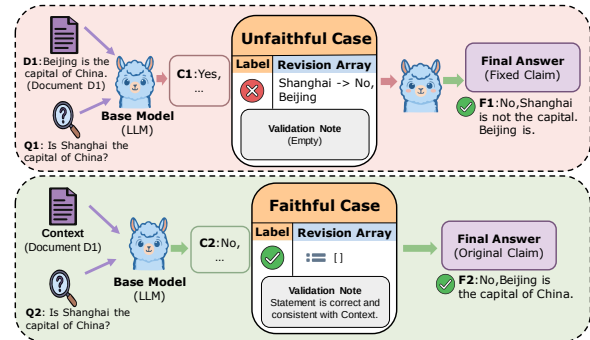


Figure 1: The examples of our RADAM.

While faithfulness hallucinations may be tolerable in general-purpose applications, they are unacceptable in high-stakes domains such as medicine and law, where even minor inaccuracies can have serious consequences (Xia et al., 2025; Pipitone and Alami, 2024). Alarmingly, as LLMs become more powerful, the severity of faithfulness hallucinations does not decrease but tends to worsen. For instance, according to the hallucination leaderboard provided by Vectara¹, the DeepSeek-R1 model exhibits a factuality hallucination rate as high as 11.3%. This highlights the urgency and importance of addressing this critical issue.

Research on mitigating faithfulness hallucinations in RAG systems remains an open challenge, with most existing work focusing primarily on hallucination detection. Current mitigation strategies can be categorized into three main approaches, namely decoding-time detection, RL-based alignment, and post-hoc correction. Each of these approaches faces its own limitations in terms of efficiency, reliability, or practicality.

Inspired by how humans revise errors—by explicitly pinpointing mistakes and formulating targeted fixes before rewriting the text—we propose

*Corresponding authors.

¹<https://github.com/vectara/hallucination-leaderboard>

a novel post-processing framework for faithfulness hallucination mitigation (Figure 1). Specifically, rather than relying on complex multi-model pipelines, we employ a single small language model (SLM) to simultaneously generate structured detection results and explicit revision suggestions, which are then passed to the base LLM for faithful response generation.

Our contributions are summarized as follows:

1. We establish a **fine-grained taxonomy** of faithfulness hallucinations and introduce a robust data synthesis pipeline via multi-model consistency filtering to construct high-quality datasets.
2. We propose **RADAM**, a lightweight post-processing framework that optimizes a single small language model to generate structured critiques—encompassing accurate detection labels, precise error spans, and explicit revision suggestions—for guided self-correction.
3. We demonstrate that **RADAM consistently achieves the lowest hallucination rates across all evaluated benchmarks**, outperforming existing mitigation methods while largely preserving response quality.

2 Related Work

2.1 Taxonomy of Hallucination

To effectively address the hallucination problem in natural language generation, establishing a comprehensive taxonomy is crucial. Previous surveys generally classify hallucinations into broad categories. For instance, Ji et al. (2023) and Ye et al. (2023) distinguish between intrinsic hallucinations (which contradict the source content) and extrinsic hallucinations (which are unverifiable by the source). Alternatively, Zhang et al. (2025) categorize them based on the locus of the conflict: input-conflicting, context-conflicting, and fact-conflicting.

2.2 Faithfulness Hallucinations Detection

There are two main approaches to detect faithfulness hallucinations in LLM-generated outputs: NLI-based post-hoc detection and LLM-based post-hoc.

NLI-based post-hoc detection frames the task as entailment classification between retrieved context and output. Vectara trains a custom NLI model (Li et al., 2024) matching GPT-4o’s performance.

Luna (Belyi et al., 2025) uses a DeBERTa-based model fine-tuned on GPT-4 annotations to balance efficiency and context handling. LettuceDetect (Kovács and Recski, 2025) improves robustness with enhanced long-document processing.

Conversely, LLM-based post-hoc detection leverages the strong reasoning and generalization capabilities of language models, shifting the focus from simple binary classification toward interpretable diagnostics. For example, RAG-HAT (Song et al., 2024) trains a specialized detection model to not only output hallucination labels but also provide detailed, human-readable descriptions of information conflicts. Similarly, FaithLens (Si et al., 2026) addresses the high inference costs typically associated with advanced LLM judges by synthesizing high-quality training data to train a compact, cost-efficient detector. FaithLens jointly outputs binary predictions and coherent explanatory rationales, allowing users to precisely localize errors without relying on opaque or expensive commercial APIs.

2.3 Faithfulness Hallucination Mitigation

To address the challenge of hallucinations in retrieval-augmented generation (RAG), various approaches have been proposed, broadly categorized into decoding-based, reinforcement learning-based, and post-processing methods.

Decoding-based approaches have proven effective in mitigating faithfulness hallucinations by encouraging the model to adhere closely to retrieved context. For instance, Lookback-Lens (Chuang et al., 2024a) mitigates hallucinations by analyzing and adjusting attention maps, while Redeeep (Sun et al., 2025) applies targeted interventions on the internal hidden states of the LLM. However, these methods often lack interpretability and require access to internal model states.

Reinforcement learning (RL)-based approaches aim to align generation with context by optimizing faithfulness-related rewards (Zhang et al., 2024). However, standard optimization paradigms introduce a fundamental dilemma. As demonstrated by Kalai et al. (2026), evaluating and optimizing LLMs strictly for structural accuracy and helpfulness inadvertently incentivizes unfaithful outputs.

Post-processing methods offer a more interpretable and modular alternative. For instance, RAG-HAT (Song et al., 2024) explores this by training a dedicated detection model alongside a separate small model that receives revision suggestions

to amend the answer. Similarly, RAGLens (Xiong et al., 2026) utilizes a Sparse Autoencoder (SAE) detector to extract hallucinated spans and returns them to the original generation model for correction. In contrast, we train a single, lightweight diagnostic model to generate both accurate detection results and detailed, structured revision data, improving both efficiency and effectiveness.

3 Hallucination Taxonomy

We inherit and further expand the taxonomy from Ji et al. (2023) and Ye et al. (2023), providing a more detailed analysis of the hallucination problem.

3.1 Categorization Framework

Our taxonomy classifies faithfulness hallucinations into five primary dimensions based on the logical relationship between the generated text and the retrieved context. The detailed categorization and task-specific manifestations are summarized in Table 1.

The five categories encompass the full spectrum of generation failures. Direct Contradiction and Internal Inconsistency represent the most explicit logical failures. Baseless Information captures external knowledge intrusion, where the model incorporates unverified details. Distortion of Context reflects semantic drift, where the model cannot fully grasp the context correctly. Finally, Compromised Rigor captures subtle nuances where the generation is not strictly false but lacks the necessary precision to be deemed fully reliable.

3.2 Task-specific Manifestations

A key contribution of this taxonomy is the isolation of task-specific error patterns. Because Summary and QA tasks impose different constraints on the generation process, the models exhibit divergent hallucination behaviors.

In Summary tasks, the model is tasked with synthesizing broad information. Consequently, Baseless Information frequently manifests not only as the introduction of unsubstantiated entities but also through subjective inferences regarding a character’s emotions or thoughts—a pattern less prevalent in rigid QA settings. Furthermore, summaries are uniquely susceptible to Compromised Rigor through a Subjective-to-Objective Shift, where a model mistakenly elevates a speculative opinion mentioned in the text to an established objective fact, or through the unauthorized Modification of

Constraints that artificially alters the validity scope of the retrieved information.

In contrast, in QA tasks, the model must extract and route precise answers. Here, Compromised Rigor predominantly takes the form of Inappropriate Granularity, where the generation is either too vague or excessively detailed to adequately address the specific query. In particular, under Distortion of Context, QA tasks exhibit two unique failure modes. The first is a fundamental misunderstanding of the context and intention of the question. Second, is the phenomenon of Unwarranted Refusal; due to overly conservative alignment training, a model may falsely claim the retrieved context is insufficient to answer the query, even when the exact evidence is present. This unhelpful rejection represents a critical erosion of system utility in RAG pipelines.

4 Method

4.1 Framework Overview

RADAM aims to mitigate hallucinations in Retrieval Augmented Generation (RAG) through a Small Language Model (SLM). As shown in Figure 2 The framework consists of three primary stages: Initial Generation, Fine-Grained Diagnosis, and Critique-Driven Revision.

Formulation and Notation: Given a user query q and a set of retrieved documents $D = \{d_1, d_2, \dots, d_k\}$, we define the context C as the composition of the query and the document set. Typically, the base language model, denoted as M_{base} , generates an answer directly conditioned on C . To prevent potential hallucinations and correct them at a fine-grained level, we introduce a lightweight diagnostic model, M_{small} . The overall workflow proceeds as follows:

1. **Initial Generation:** The base model M_{base} first generates a preliminary answer a_0 based on the query and context:

$$a_0 = M_{base}(C)$$

2. **Fine-Grained Diagnosis:** The initial answer a_0 , along with q and C , is fed into the diagnostic model M_{small} . Instead of a simple textual feedback, M_{small} is trained to output a structured JSON response consisting of three components: a boolean label $y \in \{\text{True}, \text{False}\}$, a revision array R , and a validation note v .

$$(y, R, v) = M_{small}(C, a_0)$$

Category	Summary Task	QA Task
Direct Contradiction	The generation presents direct contradiction or opposition to the retrieved document.	
Baseless Information	The response includes information which is not substantiated by or reasonably inferred from the document. Key manifestations: [B1] Introduction of unsubstantiated entities. [B2] Unsubstantiated abductive inference with unwarranted causal attribution. [B3] Subjective inference of characters’ emotions or thoughts.	The response includes information which is not substantiated by or reasonably inferred from the document. Key manifestations: [B1] Introduction of unsubstantiated entities. [B2] Unsubstantiated abductive inference with unwarranted causal attribution.
Internal Inconsistency	The generation is self-contradictory, containing conflicting statements within its own text.	
Distortion of Context	Failure to fully grasp the context, resulting in: [D1] Omission of key information or details. [D2] Alteration of key information or details.	Failure to fully grasp the context, resulting in: [D1] Omission of key information or details. [D2] Alteration of key information or details. [D3] Misunderstanding the question context and intention. [D4] Unwarranted refusal to answer or false claims of absent information (despite sufficient evidence).
Compromised Rigor	Lacks precision, potentially misleading the reader: [C1] Presenting a character’s subjective opinion as an established objective fact. [C2] Unauthorized alteration of qualifiers or scope of validity.	Lacks precision, potentially misleading the reader: [C3] Too general or too specific to adequately address the question.

Table 1: The proposed fine-grained taxonomy of faithfulness hallucinations. General errors (Direct Contradiction, Internal Inconsistency) are unified, while task-specific manifestations are detailed for Baseless Information, Distortion of Context, and Compromised Rigor.

Specifically, if $y = \text{True}$ (hallucination detected), R contains a list of specific hallucinated points, violation reasons, and targeted revision suggestions, while v is left empty. If $y = \text{False}$, R is empty, and v provides a rationale validating the correctness of a_0 .

- Critique-Driven Revision:** We employ a gating mechanism based on the diagnostic label y . If $y = \text{False}$, the framework directly outputs a_0 as the final answer a_{final} , thereby saving computational costs. Conversely, if $y = \text{True}$, the base model M_{base} is prompted to revise its initial answer conditioned on the detailed critique provided in the revision array R :

$$a_{final} = \begin{cases} a_0, & \text{if } y = \text{False} \\ M_{base}(C, a_0, R), & \text{if } y = \text{True} \end{cases}$$

Through this structured pipeline, our framework effectively localizes multiple fine-grained hallucination errors simultaneously and leverages targeted, element-wise feedback to guide the base model toward a more faithful generation.

4.2 Source Data Selection

We sample data from both QA and summarization datasets. The RAGTruth (Niu et al., 2024) dataset contains summaries and question-answers generated by various LLMs, along with fine-grained human-annotated hallucination labels; we utilize both the QA and Summary portions from this dataset to capture natural, in-the-wild hallucinations. For the QA tasks, we employ NQ-Open, HotpotQA, and Dolly (Conover et al., 2023). Specifically, the data processing for NQ-Open is consistent with Lookback-Lens (Chuang et al., 2024a), while HotpotQA follows the RAG-setting construction defined in Ragbench (Friel et al., 2024), and our usage of Dolly is strictly restricted to its closed QA split. For the summarization tasks, we utilize XSum (Narayan et al., 2018) and CNN/DailyMail (See et al., 2017), which provide news documents of varying lengths.

4.3 Structured Data Construction

To train the diagnostic model M_{small} to accurately output the structured JSON evaluations (i.e., labels, revision arrays, and validation notes), and to construct reliable test sets for evaluation, we generate

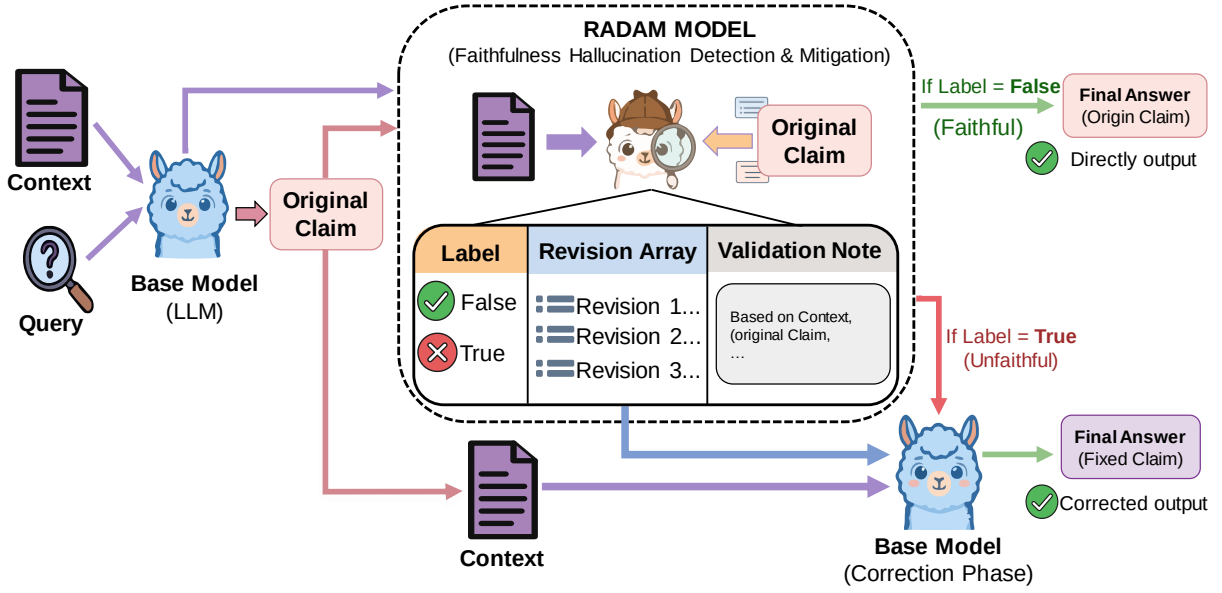


Figure 2: The illustration of our RADAM.

a suite of high-quality, fine-grained datasets. As illustrated in the data synthesis pipeline, our construction process consists of three rigorous stages:

Stage 1: Base Data Collection and Consistency Filtering. We first collect source queries and contexts from two representative RAG tasks: Question Answering (QA) and Summarization.

- *Unannotated Datasets:* For QA, we utilize NQ-Open, Dolly, and HotpotQA. For Summarization, we employ XSum and CNN/DailyMail. We use Gemini-2.5-Flash² as the base generator to produce initial answers. To ensure label accuracy, we employ a cross-model consistency filtering mechanism using GPT-5-Mini and Gemini-3-Flash as independent evaluators. An instance is retained only if both evaluators agree on the hallucination label.
- *Annotated Datasets:* To harvest high-quality natural hallucinations, we incorporate RagTruth-QA and RagTruth-Summary. We leverage their original human-annotated labels as a gold standard. We then apply the dual-evaluator consistency filtering (GPT-5-Mini³ and Gemini-3-Flash⁴) on these datasets. An

²<https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash>

³<https://developers.openai.com/api/docs/models/gpt-5-mini>

⁴<https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview>

instance is retained as a valid hallucination sample only if the judgments of both evaluators strictly align with the human ground truth, ensuring that the selected hallucinations are unambiguous and of high fidelity.

Stage 2: Targeted Hallucination Injection. While Stage 1 provides a reliable set of both truthful and naturally hallucinated responses, the distribution of fine-grained hallucination types is often imbalanced. To construct robust negative samples, we isolate the strictly non-hallucinated (faithful) instances from Stage 1. We then systematically inject targeted hallucinations into these clean answers. This injection process is guided by our fine-grained taxonomy in Section 3, ensuring comprehensive coverage of diverse hallucination patterns.

Stage 3: Final Data Generation and Quality Assurance. In the final stage, we synthesize the complete data pairs, encompassing the initial answer, the diagnostic label, the structured critique (revision array), and the validation note. We leverage Qwen3-32B(Team, 2025) to generate these detailed annotations. To maintain strict quality control, we apply a dual-stage filtering mechanism:

- *Label Verification:* We discard any generated instance where the diagnostic label predicted by Qwen3-32B contradicts the ground-truth or injected label established in Stages 1 and 2.
- *Format Validation:* Since M_{small} requires a highly structured output, we implement an au-

tomated syntax checker to filter out any data points with malformed or invalid JSON formats.

Following quality assurance, the generated instances are partitioned into training, valid and testing splits. Table 2 reports statistics of the data split.

Dataset	Train	Valid	Test
RAGTruth-QA	1,195	158	-
HotpotQA	3,356	371	660
NQ-Open	3,314	407	413
Dolly	2,025	269	258
RAGTruth-Summary	1,542	235	-
CNN/DailyMail	5,282	667	351
XSum	3,521	365	417

Table 2: Dataset statistics and partitioning details. A hyphen ("-") denotes that the corresponding dataset is excluded from the held-out test set.

4.4 Model Fine-Tuning

Pre-trained Model Backbone. After creating high-quality structured data through three stages above, we utilize Qwen3-4B-Instruct-2507 (Team, 2025) as the foundational model for our fine-tuning stage. The architecture’s 256K-token context window effectively accommodates the substantial input lengths required for analyzing multi-document contexts and comprehensive summaries. Moreover, the 4B variant offers a strategic advantage: it enables practical on-premises enterprise deployment while bypassing the Small Model Learnability Gap (Li et al., 2025) characteristic of smaller architectures.

Two-Stage Fine-Tuning. We adapt Qwen3-4B-Instruct-2507 into a specialized diagnostic model through a sequential fine-tuning approach. First, we perform supervised fine-tuning (SFT) to teach the model to map input tuples to our strict JSON taxonomy, establishing its fundamental capability to localize hallucinated spans. To further align its reasoning depth, we subsequently apply Direct Preference Optimization (DPO). By designating the comprehensive critiques from Qwen3-32B as chosen completions and the baseline 4B outputs as rejected ones, DPO explicitly penalizes both superficial critiques and incorrect diagnostic labels. This process distills the rigorous analytical capabilities of stronger models into our compact architecture without relying on complex RLHF pipelines.

Full-parameter fine-tuning is inherently memory-intensive to apply this approach. For parameter-

efficient fine-tuning, we adopt LoRA (Hu et al., 2022), which updates only small adapter layers while keeping the base model frozen. This significantly reduces computational costs and memory footprint, simultaneously preventing catastrophic forgetting during task adaptation (Bafghi et al., 2025). Reproducibility and fine-tuning details are in Appendix A.

5 Experiments

5.1 Evaluation

In-Domain Evaluation. Our in-domain evaluation leverages the strictly held-out test splits from the source datasets utilized during the training phase. The evaluation spans two representative RAG tasks: Question Answering (QA) and Summarization. Specifically, the QA test suite comprises instances from HotpotQA, Nq-Open, and Dolly, while the Summarization test suite includes CNN/DailyMail and XSum. As previously noted, the RAGTruth datasets are exclusively reserved for training and validation to supply high-quality natural hallucination signals, and are thus omitted from the testing phase. This yields a robust and diverse in-domain benchmark totaling 2,099 instances.

Out-of-Domain Evaluation. For the out-of-domain test, we select the following datasets and benchmarks:

- **FaithBench** (Bao et al., 2025), a faithfulness evaluation benchmark for summarization.
- **RagTruth++** (Blue Guardrails, 2025), a re-annotation of a subset of the RAGTruth-benchmark that spans both QA and Summarization tasks.

5.2 Hallucination Detection Experiments

The benchmark experiments evaluate the hallucination detection capabilities of our proposed framework across both in-domain and out-of-domain settings. We compare RADAM against a variety of models, including off-the-shelf small evaluators (HHEM-2.1-Open), specialized RAG evaluation frameworks (FaithLens, Raglens, Ragas), and an enterprise-level LLM (GPT-4o with FACTS CoT prompting (Jacovi et al., 2025)). Details of the baselines can be seen in Appendix B.4. In Table 3 and Table 4, all results are reported using the F1-macro metric. Bolded results indicate the best performance, and underlined results indicate the second-best performance.

Method	CNN/DailyMail			XSum			NQ-Open			HotpotQA			Dolly			Avg
	P	R	F1-M	P	R	F1-M	P	R	F1-M	P	R	F1-M	P	R	F1-M	
HHEM-2.1-Open	95.45	13.46	15.96	95.92	25.27	59.71	86.83	58.03	75.07	77.91	39.44	62.35	88.24	70.87	83.56	59.18
Gpt-4o(FACTS CoT)	98.13	67.31	83.89	100.00	78.49	86.99	<u>97.08</u>	86.01	91.32	<u>95.53</u>	86.34	91.86	95.35	96.85	96.15	<u>89.98</u>
Ragas(Gpt-4o)	78.98	79.49	81.26	83.17	90.32	87.47	78.65	72.54	77.74	82.24	66.15	76.17	85.82	90.55	87.76	81.00
FaithLens	83.01	81.41	<u>84.10</u>	92.64	81.18	<u>88.42</u>	97.52	61.14	84.01	94.85	57.14	76.76	<u>93.55</u>	<u>91.33</u>	92.63	83.68
RagLens	67.11	64.10	69.56	81.94	68.28	78.42	94.48	<u>88.60</u>	<u>92.04</u>	96.96	<u>89.13</u>	<u>93.31</u>	80.42	90.55	84.46	85.04
RADAM	85.14	<u>80.77</u>	84.92	91.11	<u>88.17</u>	90.75	96.22	92.23	94.66	94.64	93.17	94.27	89.78	96.85	<u>93.02</u>	91.93

Table 3: The in-domain test results. P and R denote Precision and Recall for the hallucinated class, respectively, while F1-M denotes Macro-F1. The Avg column represents the sample-weighted mean Macro-F1 across all tasks. This calculation applies to all subsequent tables unless otherwise specified.

Method	RagTruth++ Sum			RagTruth++ QA			FaithBench			Avg
	P	R	F1-M	P	R	F1-M	P	R	F1-M	
HHEM-2.1-Open	87.50	10.00	25.64	100.00	17.58	40.13	77.42	9.39	33.18	33.45
Gpt-4o(FACTS CoT)	97.62	29.29	42.41	88.61	42.42	55.47	86.05	21.72	43.05	45.46
Ragas(Gpt-4o)	92.45	70.00	<u>65.34</u>	<u>89.93</u>	75.76	74.66	<u>81.88</u>	47.75	56.94	61.78
FaithLens	<u>96.30</u>	37.14	47.54	82.86	52.73	60.88	78.42	29.16	46.04	49.26
RagLens	81.44	97.14	51.62	72.97	98.18	52.97	69.70	85.52	51.86	52.05
RADAM	90.00	<u>83.57</u>	70.26	84.21	<u>77.58</u>	<u>70.18</u>	74.29	<u>61.06</u>	<u>56.54</u>	<u>61.36</u>

Table 4: The Out-of-domain test results.

Analyzing the in-domain hallucination detection results (Table 3) yields the following insights: First, RADAM achieves an outstanding average F1 score of 91.93, representing a 1.95% improvement over the strongest enterprise-level baseline, GPT-4o (89.98), and a 6.89% improvement over the best specialized small-model framework, RagLens (85.04). Second, RADAM consistently dominates across diverse task types, securing the highest F1 scores on CNN/DailyMail (84.92), XSum (90.75), NQ-Open (94.66), and HotpotQA (94.27). On the Dolly dataset, RADAM remains highly competitive (93.02), ranking second only to GPT-4o. These results clearly demonstrate that with our proposed data synthesis and DPO alignment pipeline, a lightweight 4B model can effectively surpass massive proprietary LLMs on seen data distributions.

Furthermore, analyzing the out-of-domain results (Table 4) highlights the robust generalization capabilities of RADAM. First, RADAM achieves the highest performance on the RagTruth++-Sum dataset with an F1 score of 70.26. Across all out-of-domain datasets, it maintains a highly competitive average F1 score of 61.36, ranking second and performing on par with Ragas (61.78), a framework heavily reliant on the computationally expensive GPT-4o API. Second, RADAM significantly outperforms both the standard GPT-4o baseline (45.46) and all other open-source lightweight evalu-

ators (e.g., RagLens at 52.05, FaithLens at 49.26) by substantial margins. This confirms that RADAM does not merely overfit to the synthetic in-domain data, but successfully internalizes the underlying diagnostic reasoning, allowing it to generalize effectively to unseen, complex natural hallucination distributions.

5.3 Hallucination Mitigation Experiments

Method	RagTruth++ Sum	RagTruth++ QA	FaithBench	Avg
Without mitigation	40.78	41.41	41.94	41.26
FaithLens	35.19	25.29	38.89	31.86
Redeep	30.13	25.68	44.17	30.97
RagLens	29.87	25.16	<u>37.78</u>	29.45
Lookback-Lens	42.21	35.48	44.44	39.89
RADAM	28.05	24.65	36.38	28.24

Table 5: Hallucination mitigation results. The evaluation metric is the hallucination rate, where a lower score indicates better performance.

When a hallucination is detected, the RADAM framework activates the reasoning correction mechanism via its structured critiques. More details of the baselines can be seen in Appendix B.4. Based on the hallucination data in Table 5, the generation model optimized with the RADAM strategy shows a significant advantage in mitigating hallucination rates. Its average hallucination rate is reduced by 29.2% compared to the strong baseline Lookback-Lens, and it successfully surpasses the top-tier baseline RagLens across all datasets. Overall, RADAM proves to be an effective and competitive lightweight mitigation framework.

Moreover, models equipped with the RADAM framework consistently achieve significantly lower hallucination rates than their unmitigated base counterparts (reducing the average rate from 41.26 to 28.24). This substantial drop—particularly the reduction from 41.41 to 24.65 on the RagTruth++-

QA setting—clearly demonstrates the effectiveness of our critique-driven hallucination mitigation framework.

However, a lower hallucination rate does not necessarily imply higher answer quality. Our ultimate goal is to enhance the trustworthiness of large language models while also ensuring high textual quality. Therefore, we further evaluate the overall quality of the final answer a_{final} .

To assess the overall quality of the revised answers, we adopt a pairwise comparison approach: specifically, we compare the mitigated answers against the original model outputs, and employ win-rate, calculated as $\frac{\text{Win} + 0.5 \times \text{Tie}}{\text{Total}}$, as our evaluation metric.

As shown in Table 6, the lightweight model under the RADAM framework demonstrates notable performance improvements in overall answer quality compared to the unmitigated baseline. Across the evaluated datasets, RADAM achieves an average win-rate of 53.28%, setting a new benchmark with an absolute improvement of 3.28% over the original model outputs (Without mitigation). Specifically on Faithbench, RADAM achieves a remarkable win-rate of 59.72%, showcasing exceptionally strong performance in preserving and enhancing textual quality.

Compared to existing methods, our framework improves the win-rate by a massive 13.36% over Redeeep and leads the strong baseline Lookback-Lens by 11.11% on Faithbench. Notably, RADAM surpasses Lookback-Lens on both RagTruth++-Sum and Faithbench, and maintains a highly competitive win-rate (51.29%) on RagTruth++-QA. These results compellingly validate that lightweight LLMs can achieve strong performance through structured critiques, effectively balancing factual accuracy with text quality in generated content.

Method	RagTruth++ Sum	RagTruth++ QA	FaithBench	Avg
Without mitigation	50.00	50.00	50.00	50.00
FaithLens	51.04	48.58	<u>53.06</u>	50.42
Redeeep	32.01	46.06	43.61	39.92
RagLens	20.13	22.90	28.06	22.76
Lookback-Lens	<u>51.29</u>	54.84	<u>48.61</u>	<u>52.23</u>
RADAM	52.27	<u>51.29</u>	59.72	53.28

Table 6: The overall quality win-rate (%) compared to the unmitigated baseline.

5.4 Ablation Study

As shown in Tables 7 and 8, the ablation study first verifies the effectiveness of the proposed training

strategy for hallucination detection. Starting from the base model, SFT substantially improves the average Macro-F1 from 81.00 to 91.14 on the in-domain test sets and from 47.62 to 59.88 on the out-of-domain test sets. Incorporating DPO further improves the average Macro-F1 to 91.93 and 61.36, respectively. Although SFT already provides the majority of the performance gain, the consistent improvements brought by DPO, particularly on the out-of-domain test sets, indicate that preference optimization further enhances the model’s ability to distinguish hallucinated and faithful responses and improves its generalization to unseen data distributions.

As shown in Table 9, the ablation study further confirms the effectiveness of the RADAM framework’s structured diagnostic components for hallucination mitigation. Starting from the base generation model without mitigation, the system exhibits a high average hallucination rate of 41.26. Directly prompting the model to inspect and correct its own response (Self-Correct) reduces the average hallucination rate to 35.48, showing that self-correction itself provides a certain mitigation effect. However, structured diagnostic feedback leads to substantially stronger improvements. Merely alerting the model to the presence of a hallucination (Only-Label) or pinpointing its specific location (Only-Points) reduces the average hallucination rate to 33.70 and 31.39, respectively. Supplying explicit revision suggestions (Only-Suggest) yields an even stronger improvement, lowering the average rate to 30.18. These results demonstrate that external diagnostic information provides more effective guidance than relying solely on the generator’s self-correction ability, and that increasingly fine-grained feedback generally leads to better hallucination mitigation.

The training strategy also plays an important role in the final mitigation performance. RADAM with SFT achieves an average hallucination rate of 30.50, while further incorporating DPO reduces it to 28.24, corresponding to an additional absolute reduction of 2.26 points. The complete RADAM framework with SFT and DPO achieves the lowest hallucination rate across all three test sets and reduces the average hallucination rate by 13.02 points compared with the unmitigated baseline. Together, these results demonstrate that the gains of RADAM arise from both effective hallucination diagnosis and preference-aligned training. Among the individual diagnostic components, explicit revision

Method	CNN/DailyMail			XSum			NQ-Open			HotpotQA			Dolly			Avg
	P	R	F1-M	P	R	F1-M	P	R	F1-M	P	R	F1-M	P	R	F1-M	
RADAM(Base)	78.77	73.72	79.08	87.66	72.58	82.28	85.00	88.08	87.14	76.47	72.67	75.70	86.78	82.68	85.25	81.00
RADAM(SFT)	<u>83.55</u>	81.41	<u>84.38</u>	<u>90.23</u>	<u>84.41</u>	<u>88.76</u>	96.24	92.75	94.88	94.90	<u>92.55</u>	<u>93.93</u>	<u>87.14</u>	<u>96.06</u>	<u>91.07</u>	<u>91.14</u>
RADAM(SFT+DPO)	85.14	<u>80.77</u>	84.92	91.11	88.17	90.75	<u>96.22</u>	<u>92.23</u>	<u>94.66</u>	<u>94.64</u>	93.17	94.27	89.78	96.85	93.02	91.93

Table 7: The ablation study results for detection in in-domain test.

Method	RagTruth++ Sum			RagTruth++ QA			FaithBench			Avg
	P	R	F1-M	P	R	F1-M	P	R	F1-M	
RADAM(Base)	95.35	29.29	41.80	90.00	32.73	49.92	68.64	47.55	48.25	47.62
RADAM(SFT)	<u>91.23</u>	<u>74.29</u>	<u>66.48</u>	<u>85.19</u>	<u>69.70</u>	<u>67.52</u>	<u>74.01</u>	<u>54.60</u>	<u>55.96</u>	<u>59.88</u>
RADAM(SFT+DPO)	90.00	83.57	70.26	84.21	77.58	70.18	74.29	61.06	56.54	61.36

Table 8: The ablation study results for detection in out-of-domain test.

Method	RagTruth++ Sum	RagTruth++ QA	FaithBench	Avg
Without mitigation	40.78	41.41	41.94	41.26
Self-Correct	36.49	31.74	41.39	35.48
RADAM(Only-Label)	35.71	29.29	38.89	33.70
RADAM(Only-Points)	31.95	28.13	37.22	31.39
RADAM(Only-Suggest)	<u>30.78</u>	26.45	<u>36.94</u>	<u>30.18</u>
RADAM(SFT)	31.43	<u>25.81</u>	38.61	30.50
RADAM(SFT+DPO)	28.05	24.65	36.38	28.24

Table 9: The ablation study results for mitigation. The evaluation metric is the hallucination rate, where a lower score indicates better performance.

suggestions are particularly important for guiding accurate corrections, while DPO further improves the model’s ability to utilize such diagnostic signals for robust hallucination mitigation.

6 Conclusion

In this paper, we introduce an effective and structured framework, RADAM, for mitigating faithfulness hallucinations in large language models. We first establish a fine-grained taxonomy of hallucinations, under the guidance of which we synthesize high-quality training and testing datasets, employing a multi-model consistency filtering strategy to ensure data quality. We then optimize the model on these well-curated data through a two-stage training pipeline, utilizing SFT followed by DPO. In this way, RADAM can deliver state-of-the-art performance across multiple in-domain datasets and achieve comparable effectiveness to enterprise-level large models on out-of-domain datasets. Overall, our RADAM framework provides a practical solution for real-world applications.

Limitations

Despite the strong performance of RADAM in both hallucination detection and mitigation, several limitations remain. First, although we establish a fine-grained taxonomy of hallucinations, its primary utility in this work is restricted to guiding the synthesis of training data. During inference, the detector still outputs binary labels (faithful vs. hallucinated) alongside its critiques, rather than explicitly classifying the specific fine-grained subtypes, which limits the diagnostic granularity for end users. Second, the decoupled design introduces the risk of error propagation—misclassifications or flawed rationales generated by the diagnostic critic may trigger unnecessary or incorrect revisions by the base model, ultimately degrading the quality of the final output. Third, the scale of the datasets used for hallucination mitigation remains relatively limited, which may constrain the comprehensiveness of our evaluation. In addition, our mitigation experiments are currently conducted only on Llama-3-8B-Instruct, and it remains unclear whether the observed improvements generalize to other language models with different architectures, scales, and capabilities. Expanding both the mitigation datasets and the set of evaluated base models represents an important direction for future work.

Acknowledgments

The authors acknowledge the financial support from the National Natural Science Foundation of China (Grant Nos. 62673340, 62276171, 62476173, 62532007), the Shenzhen Science and Technology Program (Grant Nos. JCYJ20240813141503005, JCYJ20240813142610014, ZDCY-20250901110940006), and the Major Special Project for Philosophy and Social Sciences Research of the Ministry of Education (Grant No. 2025JZDZ010).

References

- Reza Akbarian Bafghi, Carden Bagwell, Avinash Ravichandran, Ashish Shrivastava, and Maziar Raissi. 2025. Fine tuning without catastrophic forgetting via selective low rank adaptation. *arXiv preprint arXiv:2501.15377*.
- Forrest Sheng Bao, Miaoran Li, Renyi Qu, Ge Luo, Erana Wan, Yujia Tang, Weisi Fan, Manveer Singh Tamber, Suleman Kazi, Vivek Sourabh, Mike Qi, Ruixuan Tu, Chenyu Xu, Matthew Gonzales, Ofer Mendelevitch, and Amin Ahmad. 2025. **FaithBench: A diverse hallucination benchmark for summarization by Modern LLMs**. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 448–461, Albuquerque, New Mexico. Association for Computational Linguistics.
- Masha Belyi, Robert Friel, Shuai Shao, and Atindriyo Sanyal. 2025. Luna: A lightweight evaluation model to catch language model hallucinations with high accuracy and low cost. In *Proceedings of the 31st international conference on computational linguistics: industry track*, pages 398–409.
- Blue Guardrails. 2025. **Ragtruth++ dataset**. Hugging Face.
- Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James Glass. 2024a. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. 2024b. **Dola: Decoding by contrasting layers improves factuality in large language models**. In *International Conference on Learning Representations*, volume 2024, pages 54158–54183.
- Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. **Free dolly: Introducing the world’s first truly open instruction-tuned llm**. Hugging Face.
- Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. **RAGAs: Automated evaluation of retrieval augmented generation**. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.
- Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.
- Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. **LoRA: Low-rank adaptation of large language models**. In *International Conference on Learning Representations*.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. **A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions**. *ACM Trans. Inf. Syst.*, 43(2):42:1–42:55.
- Alon Jacovi, Andrew Wang, Chris Alberti, Connie Tao, Jon Lipovetz, Kate Olszewska, Lukas Haas, Michelle Liu, Nate Keating, Adam Bloniarz, and 1 others. 2025. The facts grounding leaderboard: Benchmarking llms’ ability to ground responses to long-form input. *arXiv preprint arXiv:2501.03200*.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. **Survey of hallucination in natural language generation**. *ACM Comput. Surv.*, 55(12):248:1–248:38.
- Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2026. **Evaluating large language models for accuracy incentivizes hallucinations**. *Nature*, 653(8116):1047–1051.
- Ádám Kovács and Gábor Recski. 2025. Lettucedetect: A hallucination detection framework for rag applications. *arXiv preprint arXiv:2502.17125*.
- Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen McKeown. 2022. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1410–1421.
- Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2024. **HHEM-2.1-Open**. Hugging Face.
- Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. 2025. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25366–25394.
- Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 295–337.

- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878.
- Nicholas Pipitone and Ghita Houir Alami. 2024. [Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain](#). *Preprint*, arXiv:2408.10343.
- Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.
- Shuzheng Si, Qingyi Wang, Haozhe Zhao, Yuzhuo Bai, Guanqiao Chen, Kangyang Luo, Gang Chen, Fanchao Qi, Minjia Zhang, Baobao Chang, and Maosong Sun. 2026. [FaithLens: Detecting and explaining faithfulness hallucination](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 14068–14099, San Diego, California, United States. Association for Computational Linguistics.
- Juntong Song, Xingguang Wang, Juno Zhu, Yuanhao Wu, Xuxin Cheng, Randy Zhong, and Cheng Niu. 2024. Rag-hat: A hallucination-aware tuning pipeline for llm in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1548–1558.
- ZhongXiang Sun, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. 2025. [RedeEP: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability](#). In *The Thirteenth International Conference on Learning Representations*.
- Manveer Singh Tamber, Forrest Bao, Chenyu Xu, Ge Luo, Suleman Kazi, Minseok Bae, Miaoran Li, Ofer Mendelevitch, Renyi Qu, and Jimmy Lin. 2025. Benchmarking llm faithfulness in rag with evolving leaderboards. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 799–811.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Peng Xia, Kangyu Zhu, Haoran Li, Tianze Wang, Weijia Shi, Sheng Wang, Linjun Zhang, James Zou, and Huaxiu Yao. 2025. [MMed-RAG: Versatile multi-modal RAG system for medical vision language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Guangzhi Xiong, Zhenghao He, Bohan Liu, Sanchit Sinha, and Aidong Zhang. 2026. [Toward faithful retrieval-augmented generation with sparse autoencoders](#). In *The Fourteenth International Conference on Learning Representations*.
- Haoming Xu, Ningyuan Zhao, Liming Yang, Sendong Zhao, Shumin Deng, Mengru Wang, Bryan Hooi, Nay Oo, Huajun Chen, and Ningyu Zhang. 2025. Relearn: Unlearning via learning for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5967–5987.
- Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. [Siren's song in the AI ocean: A survey on hallucination in large language models](#). *Computational Linguistics*, 51(4):1373–1418.
- Zongmeng Zhang, Yufeng Shi, Jinhua Zhu, Wengang Zhou, Xiang Qi, peng zhang, and Houqiang Li. 2024. [Trustworthy alignment of retrieval-augmented large language models via reinforcement learning](#). In *Forty-first International Conference on Machine Learning*.

A Reproducibility and Fine-Tuning Details

A.1 Technical Reproducibility

To facilitate reproducibility and transparency, we report the hardware and software environment used for all experiments. RADAM was fine-tuned on four NVIDIA A100 GPUs, leveraging the DeepSpeed framework to optimize memory efficiency and significantly accelerate the multi-GPU training process. The experiments were carried out on a Linux server running CUDA 12.4 and PyTorch 2.10.0. Our source code available at <https://github.com/nog0/RADAM>.

A.2 Fine-Tuning Configuration

Parameter	Value
lora_target	all
lora_rank	64
lora_alpha	64
precision	bfloat16
epochs	3
batch_size	4
grad_accumulation	8
effective_batch_size	32
optimizer	AdamW
learning_rate	1×10^{-5}
lr_schedule	cosine
warmup_ratio	0.1

Table 10: The configuration of SFT fine-tuning.

Parameter	Value
lora_target	all
lora_rank	64
lora_alpha	64
precision	bfloat16
epochs	1
batch_size	4
grad_accumulation	8
effective_batch_size	32
optimizer	AdamW
learning_rate	1×10^{-6}
lr_schedule	cosine
warmup_ratio	0.1
pref_beta	0.1

Table 11: The configuration of DPO fine-tuning.

B Experimental Details

B.1 Inference Parameters

Parameter	Value
temperature	0.7
min_p	0.0
top_p	0.8
top_k	20

Table 12: Inference parameters used in our experiments, following the recommended Qwen settings.

B.2 Hallucination Detection Experimental Details

As noted in the main text, we evaluate the zero-shot generalization of RADAM on several out-of-domain benchmarks to verify its robustness across unseen contexts and tasks. Due to space constraints in the main text, we provide the detailed sample statistics and domain distributions for each out-of-domain evaluation dataset in Table 13.

Dataset	Task Type	Sample Size
Faithbench	Summary	750
RagTruth++-Sum	Summary	174
RagTruth++-QA	QA	234
Total	—	1158

Table 13: Detailed statistics of the out-of-domain evaluation datasets.

B.3 Hallucination Mitigation Experimental Details

Target Base Model (M_{base}) For our hallucination mitigation experiments, we select **Llama-3-8B-Instruct** as the primary target base language model.

Mitigation Datasets and Strategy To evaluate the efficacy of our mitigation pipeline, we construct a dedicated test bed utilizing three established hallucination benchmarks. The detailed sample statistics for the mitigation phase are provided in Table 14. Notably, during dataset curation, we observed that the standard test set for RagTruth++ contained an insufficient number of queries for a robust evaluation. To address this data scarcity, we strategically supplemented it by mixing in relevant test samples from the original RagTruth dataset, thereby ensuring a comprehensive and statistically significant evaluation suite.

Dataset	Sample Size
FaithBench	72
RagTruth++-Sum*	154
RagTruth++-QA*	155

Table 14: Detailed statistics of the datasets utilized for the hallucination mitigation experiments. *Indicates that the dataset is strategically augmented with samples from the original RagTruth test set to ensure a sufficient evaluation scale.

Evaluation Protocol Assessing the correctness of mitigated outputs requires a highly reliable oracle. Given that all three selected datasets provide high-quality human annotations, we leverage these human-verified examples to enhance the evaluation capability of large language models. Specifically, we employ **GPT-5-Mini** as our automated judge, utilizing a few-shot prompting strategy. As demonstrated in prior studies (Tamber et al., 2025), incorporating ground-truth human annotations as in-context demonstrations significantly calibrates and elevates GPT-5-Mini’s proficiency in identifying subtle factual inconsistencies, thereby ensuring a stringent and accurate assessment of RADAM’s mitigation performance. To mitigate the inherent variance of LLM-based evaluators and ensure statistical reliability, we repeat all experimental runs five times and report the averaged metrics.

B.4 Baseline Methods

This section compares the proposed RADAM framework with several recent state-of-the-art baseline methods, as detailed below:

- **HHEM-2.1-Open** (Li et al., 2024): Built on the T5 model, it targets hallucination detection by measuring the consistency between hypothesis and premise to improve content reliability. In our study, it is utilized exclusively as a baseline for hallucination detection.
- **Ragas** (Es et al., 2024): An evaluation framework that utilizes an LLM to decompose the generated answer into multiple atomic facts. It then systematically verifies each fact against the retrieved context to compute an aggregated Faithfulness Score. In our study, we employ it exclusively as a baseline for hallucination detection.
- **FaithLens** (Si et al., 2026): Primarily designed as a hallucination detection model, it distinguishes itself by providing human-readable rationales for the factual inconsistencies it identifies. Because these explicit explanations can naturally serve as corrective guidance, we employ it as a baseline not only in our detection evaluation but also in our post-processing mitigation experiments.
- **Raglens** (Xiong et al., 2026): A white-box framework that extracts intermediate hidden representations during generation and utilizes Sparse Autoencoders (SAEs) to decompose them into interpretable sparse features. It isolates hallucination-specific features via Mutual Information (MI) and employs an interpretable model (e.g., GAM) to predict factual conflicts. In our study, we employ it as a comprehensive baseline in both our detection and mitigation experiments. Before evaluation, we retrain Raglens using the same training data as our method to ensure a fair comparison.
- **Lookback-Lens** (Chuang et al., 2024a): An advanced white-box mitigation strategy that leverages the internal hidden states and attention mechanisms of the base LLM to reduce factual contradictions. Because it requires access to model weights and focuses on intervention, it is utilized strictly as a baseline for our mitigation experiments.
- **Redeep** (Sun et al., 2025): Another robust white-box intervention framework that manipulates internal model representations to dynamically suppress hallucinations during the generation process, serving as a dedicated baseline for the mitigation phase.

C Supplementary Materials for Hallucination Taxonomy

To help better understand the meaning of each hallucination type, we provide examples in Table 15.

D Detailed Case Studies of Hallucination Mitigation

To provide a more intuitive understanding of our proposed framework in practice, we presents representative qualitative examples of the hallucination mitigation pipeline, as shown in Table 16.

E Prompt

The core prompt templates used in our experiments are presented from Table 17 to Table 24.

Category	Manifestations	Retrieved Context	Hallucinated Output
Direct Contradiction	-	The company's profit increased by 7.2% year-on-year	The company's profit dropped 7.2% year over year
Baseless Information	[B1]	Beijing Library has a large collection of science fiction books, including the famous " The Three-Body Problem " series.	Beijing Library has a large number of science fiction books such as "The Three-Body Problem" series and " The Wandering Earth ".
	[B2]	The error rate dropped by 15%. The team also implemented a new data filtering pipeline.	The error rate dropped by 15% because of the new data filtering pipeline.
	[B3]	The CEO announced his resignation during the Q3 earnings call.	The CEO angrily announced his resignation during the Q3 earnings call.
Internal Inconsistency	-	-	The evaluation relies entirely on F1-macro . However, no classification metrics are used .
Distortion of Context	[D1]	To deploy the API, you need a valid key, an active subscription, and a whitelisted IP .	To deploy the API, you need a valid key .
	[D2]	The default learning rate for the reward model is set to 2e-5 .	The default learning rate for the reward model is set to 5e-2 .
	[D3]	[Q]: What are the disadvantages of RLHF? [Doc]: RLHF aligns models well but is computationally expensive.	The advantages of RLHF include better model alignment.
	[D4]	[Q]: What is the context window? [Doc]: The model supports a context window of 8K tokens .	The provided document does not mention the context window length.
Compromised Rigor	[C1]	The author argues that this algorithm might be the most efficient approach to date.	This algorithm is the most efficient approach to date.
	[C2]	The configuration is optimized for models under 10B parameters .	The configuration is optimized for all large language models.
	[C3]	[Q]: Which exact metric was used? [Doc]: We utilized FaithBench to evaluate faithfulness.	We used a benchmarking dataset to evaluate it.

Table 15: Concrete examples of different fine-grained faithfulness hallucinations across tasks. We highlight the **hallucinated text** in red and the **original context** it deviates from in blue.

Case 1: Mitigating Unwarranted Refusal in Question Answering	
User Query	how do i breed a shugabeats in my singing monsters
Retrieved Context	Passage 1: [...] you need to combine correct elements in order to breed your desired monster. Passage 2: [...] To breed the Shugabeats, you need to use the Shugabush and the Furcorn in the Shugabush Island breeding structure to get this monster. Passage 3: Generation 2 My Singing Monsters Combo Breeding. [...]
Base Output	Unable to answer based on given passages.
RADAM Critique	• Label: true (Refusal) • Violation 1 (Refusal): The document explicitly provides step-by-step instructions for breeding Shugabeats in passage 2 (Shugabush + Furcorn). The response incorrectly claims the information is absent. • Revision Suggestion: Replace the refusal with a concise summary of the breeding steps.
RADAM Revision	To breed a Shugabeats, first obtain a Shugabush Monster from the other island (it will become an egg). Then, purchase the breeding structure on Shugabush Island and combine it with a Furcorn to produce a Shugabeats.
Case 2: Mitigating Baseless Information and Distortion in Summarization	
User Query	Summarize the following news.
Retrieved Context	[...] He was taken to an area hospital as a precaution, the airline said. He passed a drug test and was discharged. [...] Alaska Airlines said it's investigating. [...]
Base Output	[...] The agent was taken to a hospital as a precaution, but was later discharged after passing a drug test. [...] The airline is investigating the incident, which is considered a serious breach of safety protocols.
RADAM Critique	• Label: true (Distortion & Baseless Information) • Violation 1 (Distortion): The phrasing “but was later discharged after...” implies a causal relationship that is not explicitly stated. • Violation 2 (Baseless): The document mentions an investigation but provides no evidence confirming it constitutes a “serious breach of safety protocols.” This is an unsubstantiated abductive inference. • Revision Suggestion: Revise the medical discharge sentence to align with the sequence of events, and remove the unsubstantiated claim about safety protocols.
RADAM Revision	[...] The agent was taken to a hospital as a precaution, later discharged after passing a drug test. [...] The airline is investigating the incident.
Case 3: Mitigating Contradiction in Instruction Generation	
User Query	how do you cook tuna steak
Retrieved Context	Passage 1 & 2: Heat the skillet [...] Add the tuna steaks. [...] Cook, uncovered, for 6 to 9 minutes [...] Passage 3: Searing tuna steaks in a hot skillet caramelizes the surfaces [...] Start with 3/4-inch-thick steaks.
Base Output	To cook tuna steak, follow these steps: [Steps 1-4 omitted for brevity]. Note: The passages do not mention the thickness of the tuna steaks, but according to passage 3, it is recommended to start with 3/4-inch-thick steaks.
RADAM Critique	• Label: true (Internal Contradiction & Direct Contradiction) • Violation 1 (Contradiction): The response incorrectly states that the thickness recommendation is absent from the passages, when passage 3 explicitly mentions 3/4-inch-thick steaks as a starting point. This is a direct contradiction of the document's content. • Revision Suggestion: Remove the contradictory phrase and revise the note to align with passage 3's explicit instructions.
RADAM Revision	To cook tuna steak, follow these steps: [Steps 1-4]. Note: Start with 3/4-inch-thick steaks.

Table 16: Comprehensive qualitative analysis of the RADAM framework.

Summary Task Prompt

You must respond based strictly on the information in the provided document. Do not incorporate any external knowledge or infer any details beyond what is given in the document.

Provide a concise summary of the following document, not omitting/altering any key details/information in the document!

#DOCUMENT#

{document}

Note: output only summary, don't include any other contents!

QA Task Prompt

You must respond based strictly on the information in the provided document.

Provide a concise answer to the following question based on the information in the provided passages. If the document doesn't contain enough information to answer the question, please reply with "Unable to answer the question!".

#QUESTION#

{question}

#DOCUMENT#

{document}

Note: output only answer, don't include any other contents!

Table 17: The prompts used to generate answer for QA and Summary tasks.

Summary Hallucination Filtering Prompt

You are an expert annotator for evaluating a complete summarization whether contains any hallucination.

#DOCUMENT#

{document}

#SUMMARIZATION#

{summary}

#HALLUCINATION DEFINITION#

1. Direct Contradiction: The summarization presents direct contraction or opposition to the document.
2. Baseless Information: The summarization includes information which is not substantiated by or reasonably inferred from the document. Inference shall only be conducted through mathematical or logical reasoning based on the existing information in the document. Unreasonable inference includes, but is not limited to:
 - Unsubstantiated Abductive Inference with Unwarranted Causal Attribution.
 - Subjective inference about characters' emotions or thoughts.
3. Internal Inconsistency: The summary is self-contradictory, containing conflicting statements within its own text.
4. Distortion of Context: The model didn't fully grasp the context correctly. This includes, but is not limited to:
 - The summary omits key information/details in the document.
 - The summary alters key information/details in the document.
5. Compromised Rigor: The summary lacks precision and may mislead or confuse the reader. This includes, but is not limited to:
 - Subjective-to-Objective Shift: Presenting a character's subjective opinion or statement as an established objective fact.
 - Modification of Constraints: Add, altering or deleting specific limiting conditions, qualifiers, or the scope of validity for the information provided in the source.

#TASK#

1. First, reason step by step whether the overall summarization is fully supported by the context.
2. If the summarization is generally completely accurate and supported based on the context, label it "consistent".
3. If the summarization contains one of the hallucination under the "hallucination definition", mark it as "hallucination" accordingly.

#OUTPUT FORMAT# (STRICT)

1. Use chain-of-thought reasoning to analyze the summary point-by-point against the document. Put your reasoning inside `<think>...</think>` tags.
2. After the reasoning, output a JSON object inside `<json>...</json>` tags.

#EXAMPLE FORMAT#

`<think>...</think>`

`<json>`

`{{`

`"label": "<consistent or hallucination>"`

`}}`

`</json>`

Table 18: The hallucination filtering prompt used by GPT-5-Mini and Gemini-3-flash for the Summary task.

QA Hallucination Filtering Prompt

You are an expert annotator for evaluating a complete response whether contains any hallucination.

#QUESTION#

{question}

#DOCUMENT#

{document}

#RESPONSE#

{response}

#HALLUCINATION DEFINITION#

1. Direct Contradiction: The response presents direct contraction or opposition to the document.
2. Baseless Infomation: The response includes information which is not substantiated by or reasonably inferred from the document. Inference shall only be conducted through mathematical or logical reasoning based on the existing information in the document. Unreasonable inference includes, but is not limited to:
 - Unsubstantiated Abductive Inference with Unwarranted Causal Attribution.
3. Internal Inconsistency: The response is self-contradictory, containing conflicting statements within its own text.
4. Distortion of Context: The model didn't fully grasp the context correctly. This includes, but is not limited to:
 - The response omits key information/details in the document.
 - The response alters key information/details in the document.
 - The response misunderstand the question context and intention.
 - The response incorrectly states that the required information is absnt from the document or declines to answer (e.g., "Unable to answer the question"), despite the document containing sufficient evidence to address the question.
5. Compromised Rigor: The response lacks precision and may mislead or confuse the reader. This includes, but is not limited to:
 - The response is either too general or too specific to address the question adequately by the retrieved documents.

#TASK#

1. First, reason step by step whether the overall response is fully supported by the context.
2. If the response is generally completely accurate and supported based on the context, label it "consistent".
3. If the response contains one of the hallucination under the "hallucination definition", mark it as "hallucination" accordingly.

#OUTPUT FORMAT# (STRICT)

1. Use chain-of-thought reasoning to analyze the response point-by-point against the document. Put your reasoning inside `<think>...</think>` tags.
2. After the reasoning, output a JSON object inside `<json>...</json>` tags.

#EXAMPLE FORMAT#

```
<think>...</think>
<json>
{{
  "label": "<consistent or hallucination>"
}}
</json>
```

Table 19: The hallucination filtering prompt used by GPT-5-Mini and Gemini-3-flash for the QA task.

Generalized Hallucination Injection Prompt (Summary Task)

You are a professional hallucination generator for text summarization. Your task is to generate a hallucinated summary that strictly meets the definition of "[Hallucination Category]".

Definition of [Hallucination Category]

[Definition of the specific hallucination category]

Generation Rules

1. Modify the clean summary to [Specific injection rule for the target manifestation] (Place the modification at a random position in the summary).
2. Keep the summary length and fluency consistent with the clean summary.
3. Do NOT add new information unless required by the specific hallucination type.
4. Output ONLY the hallucinated summary, no extra explanation or labels.

Few-Shot Examples

Note: Hallucination Reason in Example is to help you understand the hallucination point in hallucinated summary.

[Task-specific few-shot examples corresponding to the target hallucination manifestation...]

Your Task

Original Document:

{document}

Clean Summary:

{clean_summary}

Output ONLY the hallucinated summary, no extra explanation or labels.

Hallucinated Summary ([Hallucination Category]):

Generalized Hallucination Injection Prompt (QA Task)

You are a professional hallucination generator for text response. Your task is to generate a hallucinated response that strictly meets the definition of "[Hallucination Category]".

Definition of [Hallucination Category]

[Definition of the specific hallucination category]

Generation Rules

1. Modify the clean response to [Specific injection rule for the target manifestation] (Place the modification at a random position in the response).
2. Keep the response length and fluency consistent with the clean response.
3. Do NOT add new information unless required by the specific hallucination type.
4. Output ONLY the hallucinated response, no extra explanation or labels.

Few-Shot Examples

Note: Hallucination Reason in Example is to help you understand the hallucination point in hallucinated response.

[Task-specific few-shot examples corresponding to the target hallucination manifestation...]

Your Task

Question:

{question}

Original Document:

{document}

Clean Response:

{clean_response}

Output ONLY the hallucinated response, no extra explanation or labels.

Hallucinated Response ([Hallucination Category]):

Table 20: The generalized prompts used to inject specific hallucination manifestations into the Summary and QA datasets. Placeholders such as [Hallucination Category] and specific injection rules are dynamically replaced based on the targeted error type during data synthesis.

Summary Critic Prompt

You are an expert for evaluating whether a complete summary contains any hallucination, and for proposing targeted revision suggestions for each identified hallucination point when hallucination exists.

#DOCUMENT#

{document}

#SUMMARY#

{summary}

#HALLUCINATION DEFINITION#

1. Direct Contradiction: The summarization presents direct contraction or opposition to the document.
2. Baseless Information: The summarization includes information which is not substantiated by or reasonably inferred from the document. Inference shall only be conducted through mathematical or logical reasoning based on the existing information in the document. Unreasonable inference includes, but is not limited to:
 - Unsubstantiated Abductive Inference with Unwarranted Causal Attribution.
 - Subjective inference about characters' emotions or thoughts.
3. Internal Inconsistency: The summary is self-contradictory, containing conflicting statements within its own text.
4. Distortion of Context: The model didn't fully grasp the context correctly. This includes, but is not limited to:
 - The summary omits key information/details in the document.
 - The summary alters key information/details in the document.
5. Compromised Rigor: The summary lacks precision and may mislead or confuse the reader. This includes, but is not limited to:
 - Subjective-to-Objective Shift: Presenting a character's subjective opinion or statement as an established objective fact.
 - Modification of Constraints: Add, altering or deleting specific limiting conditions, qualifiers, or the scope of validity for the information provided in the source.

#OUTPUT FORMAT# (STRICT)

1. Output ONLY a JSON object wrapped in `<json>...</json>`.

#JSON FIELD RULES#

label: Boolean. true if hallucination exists, false if no hallucination.

revision: Array of objects. Only filled when label is true; empty array ([]) when label is false. Each object in the array must contain three sub-fields:

- hallucination_point: Specific content in the summary that constitutes a hallucination.
 - violation_reason: Give the detailed reason why it has hallucination, don't need to give the hallucination type.
 - revision_suggestion: Targeted modification advice for this hallucination point, which should align with the document content.
- validation_note: String. Only filled when label is false; empty string (") when label is true. This field needs to clearly explain why the summary has no hallucination, including verification of consistency between summary content and document, compliance with inference rules, absence of internal contradictions, etc.

If hallucination exists:

```
<json>{{
  "label": true,
  "revision": [{{"hallucination_point": "<Specific hallucinated content in the summary>","violation_reason":"<reason>","revision_suggestion":"<Targeted modification advice for this point>"}]},
  "validation_note": ""
}&lt;/json>
```

If no hallucination exists:

```
<json>{{
  "label": false,
  "revision": [],
  "validation_note": "<Detailed explanation of why the summary has no hallucination>"
}&lt;/json>
```

Table 21: The prompt used by Qwen3-32B and RADAM for the Summary task.

QA Critic Prompt

You are an expert for evaluating whether a complete response contains any hallucination, and for proposing targeted revision suggestions for each identified hallucination point when hallucination exists.

#QUESTION#

{question}

#DOCUMENT#

{document}

#RESPONSE#

{response}

#HALLUCINATION DEFINITION#

1. Direct Contradiction: The response presents direct contraction or opposition to the document.
2. Baseless Information: The response includes information which is not substantiated by or reasonably inferred from the document. Inference shall only be conducted through mathematical or logical reasoning based on the existing information in the document. Unreasonable inference includes, but is not limited to:
 - Unsubstantiated Abductive Inference with Unwarranted Causal Attribution.
3. Internal Inconsistency: The response is self-contradictory, containing conflicting statements within its own text.
4. Distortion of Context: The model didn't fully grasp the context correctly. This includes, but is not limited to:
 - The response omits key information/details in the document.
 - The response alters key information/details in the document.
 - The response misunderstand the question context and intention.
 - The response incorrectly states that the required information is absent from the document or declines to answer (e.g., "Unable to answer the question"), despite the document containing sufficient evidence to address the question.
5. Compromised Rigor: The response lacks precision and may mislead or confuse the reader. This includes, but is not limited to:
 - The response is either too general or too specific to address the question adequately by the retrieved documents.

#OUTPUT FORMAT# (STRICT)

1. Output ONLY a JSON object wrapped in `<json>...</json>`.

#JSON FIELD RULES#

label: Boolean. true if hallucination exists, false if no hallucination.

revision: Array of objects. Only filled when label is true; empty array ([]) when label is false. Each object in the array must contain three sub-fields:

- hallucination_point: Specific content in the response that constitutes a hallucination.
 - violation_reason: Give the detailed reason why it has hallucination, don't need to give the hallucination type.
 - revision_suggestion: Targeted modification advice for this hallucination point, which should align with the document content.
- validation_note: String. Only filled when label is false; empty string ("") when label is true. This field needs to clearly explain why the summary has no hallucination, including verification of consistency between response content and document, compliance with inference rules, absence of internal contradictions, etc.

If hallucination exists:

```
<json>{{
  "label": true,
  "revision": [{{"hallucination_point": "<Specific hallucinated content in the summary>","violation_reason":"<reason>","revision_suggestion":"<Targeted modification advice for this point>"}]},
  "validation_note": ""
}}</json>
```

If no hallucination exists:

```
<json>{{
  "label": false,
  "revision": [],
  "validation_note": "<Detailed explanation of why the summary has no hallucination>"
}}</json>
```

Table 22: The prompt used by Qwen3-32B and RADAM for the QA task.

Summary Revision Prompt

Given the following components, generate a concise and accurate summary based on the document.
You can use the Revisions section to correct any errors in the response, ensuring the summary strictly reflects the document.
The summary should capture the key points and main ideas while avoiding unnecessary details.
Do not introduce any external knowledge beyond the document.
Output only the final summary.
#Document#: {document}
#Response#: {response}
#Revisions#: {revisions}
Output:

QA Revision Prompt

Given the following components, output the most accurate and factually correct answer to the question based on the document.
You can use the Revisions section to correct any errors in the response, ensuring the answer strictly reflects the document.
Do not introduce any external knowledge beyond the document.
Output only the final answer.
#Document#: {document}
#Question#: {question}
#Response#: {response}
#Revisions#: {revisions}
Output:

Table 23: The prompts used for the base model to generate the final revised output, incorporating the targeted revisions provided by the Critic model.

Pairwise Comparison Prompt (Summary Task)

You are an expert evaluator. You will be provided with a Reference Document and two AI-generated Summaries (Summary A and Summary B).
Your task is to evaluate which summary is of higher quality based strictly on the Reference Document.
Evaluation Criteria:
1. Faithfulness: The summary must not contain any hallucinations or fabricate information outside the Reference Document.
2. Coverage: The summary should capture the main points and key information of the document without missing crucial details.
3. Conciseness & Clarity: The summary should be concise, coherent, and easy to read.
[Reference Document]
{document}
[Summary A]
{summary_a}
[Summary B]
{summary_b}
Think step-by-step and output your final verdict in valid JSON format containing exactly two keys: 'reasoning' (a string explaining your evaluation) and 'choice' (a string which must be exactly 'A', 'B', or 'Tie' if they are of equal quality).

Pairwise Comparison Prompt (QA Task)

You are an expert evaluator. You will be provided with a Reference Document, a User Question, and two AI-generated Answers (Answer A and Answer B).
Your task is to evaluate which answer is of higher quality based strictly on the Reference Document.
Evaluation Criteria:
1. Faithfulness: The answer must not contain any hallucinations or fabricate information outside the Reference Document.
2. Accuracy: The answer should correctly and fully address the User Question using only the provided document.
3. Clarity: The answer should be logically structured and easy to read.
[Reference Document]
{document}
[User Question]
{question}
[Answer A]
{answer_a}
[Answer B]
{answer_b}
Think step-by-step and output your final verdict in valid JSON format containing exactly two keys: 'reasoning' (a string explaining your evaluation) and 'choice' (a string which must be exactly 'A', 'B', or 'Tie' if they are of equal quality).

Table 24: The prompts used by the LLM evaluator to conduct pairwise quality comparisons between the baseline and our proposed method for both Summary and QA tasks.