LLM-Generated Text May Harm Your Retrieval! A Robust Detection Strategy for Retrieval-Augmented Generation

Zhaoheng Huang, Yutao Zhu*, Ji-Rong Wen, and Zhicheng Dou*
[1]Gaoling School of Artificial Intelligence, Renmin University of China
{huangzh,ytzhu,dou}@ruc.edu.cn

Abstract

Retrieval-augmented generation (RAG) effectively enhances the accuracy and timeliness of large language models (LLMs) by incorporating external knowledge retrieved from external sources. However, with the increasing prevalence of LLM-generated content, external corpora used by RAG systems may become contaminated with LLM-generated texts. Such contamination compromises the reliability and quality of retrieved results, ultimately leading to a degradation in RAG performance, and raises concerns about the diminishing presence of human texts and the "Spiral of Silence" effect. A natural solution is to incorporate LLM text detectors into the RAG pipeline to filter out LLM-generated texts from the retrieved results. However, their effective use in RAG remains under-explored. In this paper, we explore the usage paradigms of LLM text detectors for RAG and highlight key limitations of off-the-shelf or directly fine-tuned detectors. To this end, we propose a RAG-aware data augmentation strategy that aligns detector training with realistic contamination patterns. Our approach synthesizes training data from both LLM and human texts under diverse generation modes. Experiments show that our method mitigates performance degradation and improves the long-term stability of RAG systems.

1 Introduction

Large language models (LLMs) have achieved strong performance across many natural language tasks (DeepSeek-AI et al., 2025; Hurst et al., 2024), yet they can still produce hallucinated or outdated content (Huang et al., 2023). Retrieval-augmented generation (RAG) mitigates this issue by providing externally retrieved relevant texts during generation, but recent studies show that retrievers tend to favor LLM-generated texts in the corpus (Yang et al., 2025; Dai et al., 2023). As LLM text increasingly appears in the web and is subsequently indexed by search systems, retrieval corpora gradually accumulate synthetic content over time. This iterative shift in the indexed corpus leads retrieval results to become increasingly dominated by LLM texts, resulting in performance degradation and the "Spiral of Silence" phenomenon, where human-written texts are increasingly absent from top search results, reducing their visibility and influence (Chen et al., 2024b; Noelle-Neumann, 1974).

In the face of the growing prevalence of LLM texts, recent work has developed LLM text detectors to distinguish human-written from LLM-generated texts (Guo et al., 2023; Solaiman et al., 2019). A natural solution is to integrate such detectors into the RAG pipeline to filter synthetic passages before generation. However, their effective deployment under evolving RAG corpora remains under-explored. We observe that both off-the-shelf detectors and detectors fine-tuned on static RAG corpora fail to prevent the long-term accumulation of LLM texts in retrieval results. As generated content is repeatedly mixed, paraphrased, and re-used across iterations, detection performance degrades, allowing hallucinated or inaccurate texts to persist and propagate through retrieval and generation. This trend is consistent with prior findings that iteratively paraphrased or style-shifted texts are increasingly difficult to detect (Koike et al., 2024; Krishna et al., 2023).

To address this limitation, we investigate the **impact of LLM text detectors on RAG systems under evolving corpora**. To simulate realistic corpus contamination patterns, we first analyze how LLM texts are produced and introduced into retrieval corpora in RAG scenarios. Observing that LLM text generation in RAG is primarily conditioned on the input query and retrieved passages, we abstract the generation process along these dimensions and identify four representative generation modes: Di-

*Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31973–31988

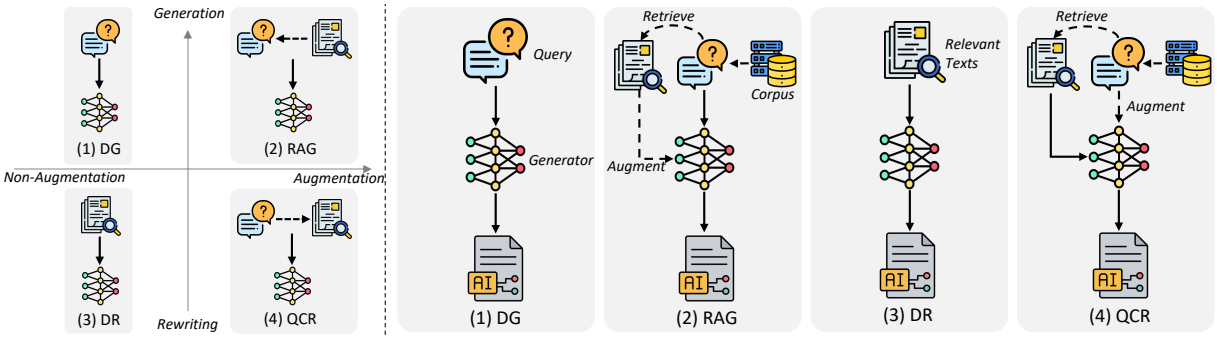

Figure 1: Four representative generation methods for producing and incorporating LLM texts into the RAG corpus.

rect Generation (DG), Retrieval-Augmented Generation (RAG), Direct Rewriting (DR), and Question-Centric Rewriting (QCR). These modes capture common ways in which LLM text enters and propagates in retrieval corpora. Details are provided in Section 3.1 and illustrated in Figure 1.

Then, building on prior work (Chen et al., 2024b), we adopt an iterative RAG framework integrated with an LLM text detector to study system behaviour under progressive corpus evolution. As shown in Figure 2, the system iteratively retrieves passages from a mixed corpus of human and LLM texts, filters detected LLM texts, and generates responses using the remaining passages as context. To simulate realistic evolution, newly generated texts are synthesized by uniformly sampling one of the four generation modes conditioned on the query and retrieved passages, and index them back into the corpus. Repeating this process over multiple iterations mimics the accumulation of LLM-generated content and its long-term impact on RAG performance (Chen et al., 2024b).

To address the limitations of existing detectors in RAG scenarios, we revisit detector training from a data-centric perspective. Prior work has shown that the effectiveness and generalization of detectors are fundamentally dependent on careful dataset construction (Wu et al., 2025; Xu et al., 2024). Motivated by this observation, we introduce a **RAG-aware data augmentation strategy** that adapts detectors to evolving RAG corpora by aligning training data with realistic contamination patterns. Our approach iteratively synthesizes training samples from mixtures of human-written and LLM-generated texts under diverse generation modes. As a result, detectors trained with this strategy become more robust to corpus evolution and more reliable in filtering LLM texts in search results.

We evaluate our approach on open-domain question answering (ODQA) benchmarks (Chen et al., 2024b), a widely used testbed for RAG systems (Wang et al., 2025; Pan et al., 2023; Zhu et al., 2025). Results show that detectors trained with RAG-aware augmented data consistently outperform off-the-shelf and directly fine-tuned baselines, effectively mitigating long-term performance degradation and alleviating the "Spiral of Silence" effect. Further mechanism analysis reveals that appropriately staged text synthesis is essential for achieving robust detection under evolving corpus contamination in the RAG system.

Our contributions are summarized as follows:

- We systematically study LLM-generated text contamination in RAG systems under realistic corpus evolution, and analyze how four representative generation modes contribute to long-term performance degradation.
- We introduce an iterative RAG evaluation framework with integrated LLM text detection, enabling a comprehensive analysis of detector usage paradigms. Our results show that both off-the-shelf and directly fine-tuned detectors fail to prevent degradation under evolving corpora.
- We present a RAG-aware data augmentation strategy that synthesizes mixtures of human-written and LLM-generated texts to improve detector robustness. Detectors trained under this paradigm better adapt to corpus evolution and effectively stabilize RAG performance over iterations.

2 Related Work

Impact of LLM Text on Information Systems The growing influx of LLM-generated text into public corpora has raised widespread concerns about the quality and reliability of information systems. Recent studies (Dai et al., 2024; Peng et al., 2024; Zhou et al., 2024) reveal that neural retrievers tend to assign higher relevance scores

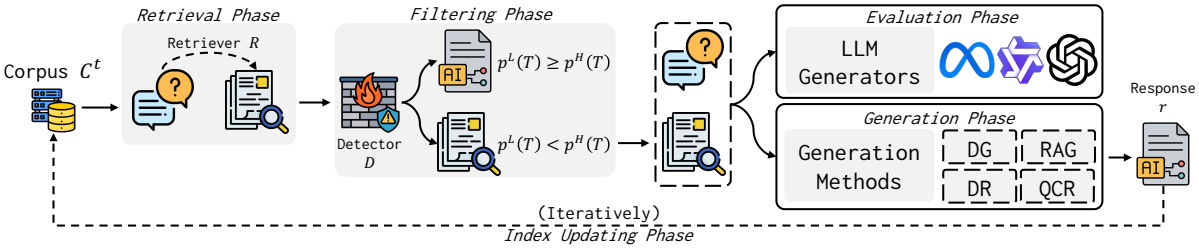

Figure 2: The simulation pipeline of the iterative RAG framework.

to LLM texts than to human texts, a phenomenon defined as "source bias". This bias leads to a higher proportion of LLM texts presence in retrieval results, potentially exposing users to inaccurate or misleading information due to the inherent hallucinations of LLMs (Huang et al., 2023). Follow-up research (Chen et al., 2024b) demonstrates that source bias is prevalent in real-world RAG systems, where the dominance of LLM texts in retrieval results diminishes the visibility of human texts, reflecting a digital "Spiral of Silence" effect (Noelle-Neumann, 1974; Yu et al., 2026; Chen et al., 2026). In this paper, we focus on the use of LLM text detectors in mitigating RAG performance degradation caused by the accumulation of LLM-generated texts.

LLM-Generated Text Detection Existing detection methods generally fall into two categories. **Metric-based** approaches identify LLM text by exploiting generative features of LLMs, such as likelihood (Gehrmann et al., 2019; Ippolito et al., 2020), token-level perturbation sensitivity (Mitchell et al., 2023; Bao et al., 2024), or cross-LLM consistency (Hans et al., 2024; Chen et al., 2025; Huang et al., 2025) without additional training data. **Training-based** approaches fine-tune pre-trained language models on paired LLM and human texts (Solaiman et al., 2019), leveraging training objectives (Guo et al., 2023; Tian et al., 2024) and model architectures (Huang et al., 2024) to further improve detection accuracy. Their effectiveness critically depends on the quality and coverage of the training data, as detectors primarily learn distributional differences between human and LLM texts. These methods are lightweight and incur minimal latency, making them suitable for real-world deployment. In this work, we focus on training-based detectors and examine how training data distributions interact with evolving RAG corpora, and how data construction affects long-term detection robustness.

3 Methodology

3.1 Problem Formulation

Given a dataset Q, a query $q \in Q$, and a corpus C^t at iteration t, a RAG system f generates a response r for q. The standard RAG system includes a retriever R and an LLM generator g from a set of LLMs G. We further add a detector D that filters LLM texts from retrieved candidates to mitigate performance degradation over corpus evolution. Specifically, the retriever R first retrieves and re-ranks the top-k relevant texts from the corpus C^t for query q, forming the candidate set $P_q^t = R(q, C^t)$. The detector D filters out texts in P_q^t predicted as LLM-generated, yielding $\tilde{P}_q^t = D(P_q^t)$. We then select the top-k texts from $\tilde{P}_q^t$ as the final context $S_q^t = \text{Top}_k(q, \tilde{P}_q^t)$.

To model corpus evolution, we uniformly sample one of four generation methods M to produce a response r, which is then indexed to the corpus (Figure 1), capturing common LLM text accumulation patterns in real-world RAG systems, conditioned on user queries and retrieved texts: **(1) Direct Generation (DG).** Generate r directly from the query: $r = g_{\text{DG}}(q)$, mimicking standalone LLM usage (Wu et al., 2024). **(2) Retrieval-Augmented Generation (RAG).** Generate r from the query and retrieved context: $r = g_{\text{RAG}}(q, S_q^t)$, matching standard RAG-based systems and AI-assisted search (Huang and Huang, 2024). **(3) Direct Rewriting (DR).** Rewrite a randomly sampled passage P_i while preserving its meaning: $r = g_{\text{DR}}(P_i)$. This approach is commonly used for paraphrasing or to evade plagiarism detection systems (Meyer et al., 2023; Kim et al., 2024). **(4) Question-Centric Rewriting (QCR).** Rewrite a sampled passage P_i conditioned on the question (Raheja et al., 2023; Shu et al., 2024), aligning the rewritten text with the question's intent: $r = g_{\text{QCR}}(q, P_i)$. Prompts for all methods are provided in Appendix A.

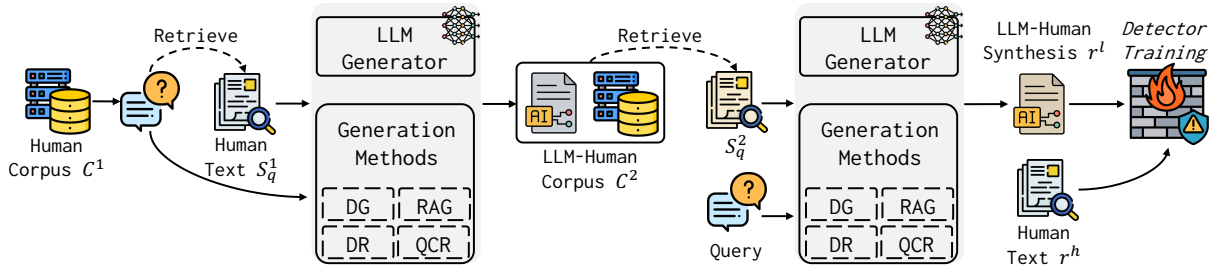

Figure 3: The proposed data augmentation strategy RAD for existing text detectors. The LLM texts are mixed from various generation methods, and synthesized from both LLM and human texts.

3.2 Pipeline Construction

Following prior work (Chen et al., 2024b), we simulate corpus evolution over T iterations. The initial corpus C^1 contains only human texts, and each iteration introduces newly generated LLM texts. As illustrated in Figure 2, each iteration consists of the following five phases:

(1) Retrieval phase. Given a query q, we retrieve and re-rank texts P_q^t from the current corpus C^t.

(2) Filtering phase. An LLM text detector D is applied to remove retrieved texts predicted as LLM-generated, and the remaining top-k retrieved texts are used as the context S_q^t.

(3) Evaluation phase. RAG responses are generated for each query q using the filtered context S_q^t across all LLM generators G, and both generation quality and detector performance are evaluated.

(4) Generation phase. For each query q, new texts are generated using a randomly selected generation method m conditioned on the query and retrieved context. The resulting set $\tilde{C} = \{r \mid r = g_m(q, S_q^t), g \in G, q \in Q\}$ is post-processed to remove identifiable LLM patterns (details are shown in Appendix B).

(5) Index updating phase. The newly generated texts are indexed into the corpus to form $C^{t+1} = C^t \cup \tilde{C}$, enabling their retrieval in subsequent iterations.

We repeat steps (1) to (5) until the maximum iteration T. Pseudocode for the entire pipeline is provided in Appendix C.

3.3 Usage Paradigms of Detectors in RAG

We first examine two conventional paradigms for incorporating LLM text detectors into RAG systems, and then introduce our data augmentation strategy tailored to corpus evolution.

Off-the-shelf detection (OTS-Det). This straightforward paradigm directly applies pre-trained detectors to filter retrieved texts. While simple to deploy, these detectors are not trained for RAG settings and tend to produce a high false positive rate, incorrectly filtering out human texts and thereby degrading retrieval quality.

Downstream fine-tuned detection (FT-Det). A more adaptive paradigm fine-tunes detectors on labeled pairs of human and LLM texts. Following prior work (Tian et al., 2024; Guo et al., 2023), we construct supervised training data by sampling queries Q_{train} from a large-scale RAG training set, retrieving human texts from the initial corpus C^1 using BM25 (Robertson et al., 1994), and pairing them with responses generated by LLaMA3.1-8B-Instruct, yielding LLM-human pairs $\{r^l, r^h\}$ as:

$$S_q^1 = \{\text{Top}_{k=5}(\text{BM25}(q, C^1)) \mid q \in Q_{\text{train}}\},$$
$$r^h = \{r_q^h \mid r_q^h = \text{Random}(S_q^1), q \in Q_{\text{train}}\},$$
$$r^l = \{g(q) \mid q \in Q_{\text{train}}\}.$$

However, this paradigm assumes a static corpus and generates LLM texts solely conditioned on the query, resulting in a training distribution that fails to capture the mixture and evolution of human and LLM texts in realistic RAG systems.

Data Augmentation Strategy for RAG Detection (RAD). Existing directly fine-tuned detectors assume that LLM and human texts are obtained in parallel and generated solely from queries (Guo et al., 2023; He et al., 2024; Hu et al., 2023). This overlooks a key challenge in RAG systems, where LLM texts are often synthesized from retrieved texts that may themselves be a mixture of human and LLM-generated texts. To address this mismatch, we adopt a **RAG-aware data augmentation approach** (RAD) that aligns detector training with realistic corpus evolution in RAG systems. The key motivation of RAD is to expose detectors during training to the same iterative contamination patterns encountered at deployment. As illustrated in Figure 3, we first collect human texts r^h as described above. For each training query $q \in Q_{\text{train}}$,

Metric	Acc@5					EM					EM_{llm}					Human@20
Dataset	NQ	WQ	TQ	PQ	Avg.	NQ	WQ	TQ	PQ	Avg.	NQ	WQ	TQ	PQ	Avg.	Avg.
Loop 1 (initial stage)																
***w/o** D*	71.5	64.5	71.0	55.5	**65.6**	63.8	60.8	78.0	55.0	**64.4**	39.0	39.2	63.2	36.7	**44.5**	**100.0**
OTS-Det	69.0	62.5	68.5	54.5	63.6$_{\nabla 3.0\%}$	62.3	58.8	74.2	54.1	62.4$_{\nabla 3.1\%}$	37.7	39.0	62.9	34.3	43.5$_{\nabla 2.2\%}$	**100.0**
FT-Det	71.5	64.5	71.0	55.5	**65.6**$_{\Delta 0.0\%}$	63.3	60.7	77.5	54.8	64.1$_{\nabla 0.5\%}$	38.2	39.5	64.0	35.2	44.2$_{\nabla 0.7\%}$	**100.0**
RAD (ours)	71.5	64.5	71.0	55.5	**65.6**$_{\Delta 0.0\%}$	63.5	60.7	77.8	54.8	64.2$_{\nabla 0.3\%}$	38.6	39.2	63.7	36.3	**44.5**$_{\Delta 0.0\%}$	**100.0**
Loop 2 (early stage)																
***w/o** D*	73.5	65.5	72.5	57.0	**67.1**	60.7	59.8	77.8	55.3	63.4	36.2	37.8	63.8	35.2	43.3	89.9
OTS-Det	72.5	65.5	72.5	57.0	66.9$_{\nabla 0.3\%}$	61.3	58.6	73.3	53.5	61.7$_{\nabla 2.7\%}$	36.0	37.2	63.6	35.0	43.0$_{\nabla 0.7\%}$	91.8
FT-Det	72.5	64.5	71.5	55.0	65.9$_{\nabla 1.8\%}$	63.7	60.0	75.5	54.3	63.4$_{\Delta 0.0\%}$	37.5	38.3	65.2	37.2	**44.6**$_{\Delta 3.0\%}$	97.5
RAD (ours)	72.0	64.5	71.5	55.0	65.8$_{\nabla 1.9\%}$	64.0	60.3	76.7	54.7	**63.9**$_{\Delta 0.8\%}$	37.7	38.7	65.0	37.1	**44.6**$_{\Delta 3.0\%}$	**98.9**
Loop 5 (mid-stage)																
***w/o** D*	70.5	65.0	71.0	55.0	**65.4**	59.8	58.8	75.7	54.8	62.3	35.5	36.0	64.0	34.8	42.6	57.2
OTS-Det	70.5	63.0	70.0	52.5	64.0$_{\nabla 2.1\%}$	60.6	58.6	71.9	51.4	60.6$_{\nabla 2.7\%}$	35.5	35.1	63.7	33.9	42.1$_{\nabla 1.2\%}$	63.9
FT-Det	71.0	64.0	70.5	54.0	64.9$_{\nabla 0.8\%}$	63.0	60.0	74.7	52.8	62.6$_{\Delta 0.5\%}$	34.7	38.8	64.0	36.3	43.5$_{\Delta 2.1\%}$	89.2
RAD (ours)	71.5	64.5	71.0	54.5	**65.4**$_{\Delta 0.0\%}$	63.7	60.5	76.5	54.4	**63.8**$_{\Delta 2.4\%}$	37.5	38.7	64.7	36.6	**44.4**$_{\Delta 4.2\%}$	**97.3**
Loop 10 (late stage)																
***w/o** D*	67.0	63.0	70.5	54.5	63.8	58.3	57.0	73.7	52.7	60.4	33.5	34.7	63.0	32.3	40.9	16.3
OTS-Det	67.5	63.0	70.0	52.0	63.1$_{\nabla 1.1\%}$	58.5	58.0	71.2	50.1	59.5$_{\nabla 1.5\%}$	33.1	34.9	63.2	31.7	40.7$_{\nabla 0.5\%}$	56.4
FT-Det	69.5	64.0	71.0	54.5	64.8$_{\Delta 1.6\%}$	59.0	58.5	73.2	53.5	61.1$_{\Delta 1.2\%}$	33.8	36.5	62.8	34.3	41.9$_{\Delta 2.4\%}$	73.7
RAD (ours)	71.5	64.5	71.0	55.0	**65.5**$_{\Delta 2.7\%}$	63.0	60.2	76.1	54.3	**63.4**$_{\Delta 5.0\%}$	36.6	38.5	63.8	36.4	**43.8**$_{\Delta 7.1\%}$	**96.7**

Table 1: Performance of RAG with the HC3 detector under various usage paradigms across four stages of corpus evolution. The best average performance within each iteration is **bold**.

we generate LLM texts using generation methods sampled uniformly from M, conditioned on both the query and its retrieved texts S_q^1:

$$r^1 = \{r \mid r = g_m(q, S_q^1), m \in M\}. \quad (1)$$

We then augment the initial corpus with these generated texts to form a mixed corpus $C^2 = C^1 \cup r^1$. New LLM texts are then generated from the query and retrieved texts S_q^2 from C^2:

$$S_q^2 = \{\text{Top}_k(\text{BM25}(q, C^2) \mid q \in Q_{\text{train}}\}, \quad (2)$$
$$r^l = r^2 = \{r \mid r = g_m(q, S_q^2), m \in M\}.$$

As a result, the augmented LLM texts are synthesized under diverse generation modes and mixed LLM-human texts, better reflecting the contamination patterns of evolving RAG corpora. This data-level design makes RAD detector-agnostic. In this work, we primarily apply RAD to the QA-oriented, training-based detector HC3 (Guo et al., 2023), which jointly takes the query and candidate retrieved text as input and is well suited for ODQA tasks in RAG pipelines. Other detector types are evaluated in Section 4.4.

3.4 Implementation Details

We set the maximum number of iterations to $T = 10$. For each query, we retrieve and rerank the top 100 relevant texts, and keep the top $k = 5$ relevant texts as RAG context after filtering. We use three representative LLMs as generators: LLaMA3.1-8B-Instruct (AI@Meta, 2024), Qwen2.5-14B-Instruct (Yang et al., 2024), and GPT-4o-mini (Hurst et al., 2024). Additional details are provided in Appendix D.

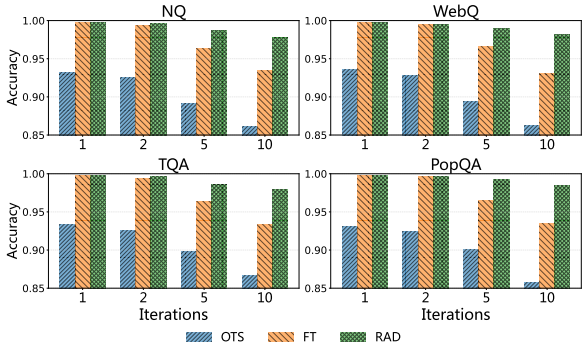

Figure 4: Detection accuracy of the HC3 detector with various usage paradigms.

4 Experiment and Analysis

4.1 Datasets and Evaluation Metrics

Datasets. Building on prior work on RAG systems (Chen et al., 2024b), we study four com-

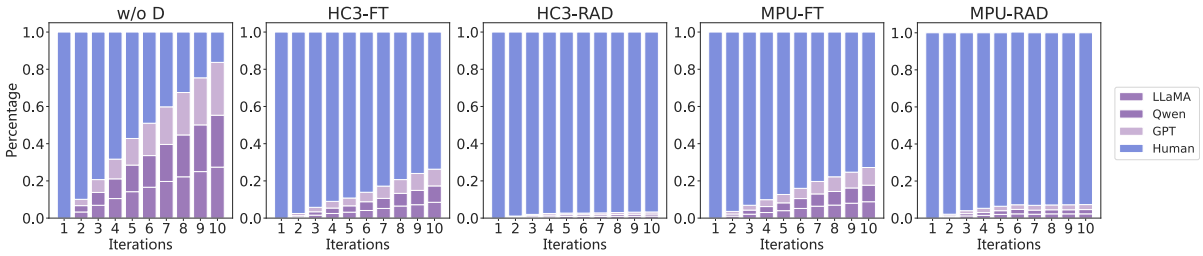

Figure 5: Proportion of human- and LLM-written texts in the top-20 retrieved results $\tilde{P}_q^t$ across iterations.

monly used ODQA datasets: **NQ** (Kwiatkowski et al., 2019), **WebQ** (Berant et al., 2013), **TriviaQA** (Joshi et al., 2017), and **PopQA** (Mallen et al., 2023). Following prior iterative RAG settings (Chen et al., 2024b), we randomly select 200 samples from each test set and form the dataset Q.

Metrics. We evaluate the system from four perspectives: retrieval performance, generation quality, detector accuracy, and the "Spiral of Silence" effect. **Retrieval performance** is measured by Acc@5 (Karpukhin et al., 2020), which evaluates whether the top-5 retrieved passages in $\tilde{S}_q^t$ contain the correct answer. **Generation quality** is evaluated using exact match (EM) (Chen et al., 2024a) and EM_{llm} (Chen et al., 2024b), where a strong LLM (GPT-5-mini) judges whether the generated response contains and supports the answer. **Detector accuracy** is measured as the classification performance on the top-100 re-ranked search results. **The "Spiral of Silence" effect** is evaluated by Human@20 (Chen et al., 2024b), which computes the proportion of human texts in top-20 filtered results.

4.2 Experimental Results

We evaluate RAG performance at four representative stages of corpus evolution: **Loop 1** (initial), **Loop 2** (early), **Loop 5** (mid), and **Loop 10** (late). Table 1 and Figure 4 report results under a representative RAG configuration with BM25 and BGE-reranker, where responses are generated using all three LLMs G. We have the following findings:

LLM text contamination progressively degrades RAG performance. Without detectors (***w/o*** D), incorporating LLM texts into the corpora leads to two distinct effects. At early stages (Loop 1 to Loop 2), a small amount of LLM texts can temporarily improve retrieval by supplementing missing answers in the corpus. However, this benefit does not translate into better generation quality, as generation performance already declines in early iterations, indicating that retrieved LLM texts often introduce noisy or hallucinated content. As corpus evolution proceeds, LLM texts increasingly dominate retrieval results, causing the proportion of human-written texts in search results to sharply drop (from 100% to 16.3%). This shift results in sustained degradation of both retrieval and generation performance, exhibiting a typical "garbage in, garbage out" pattern and reflecting a "Spiral of Silence" effect in RAG systems.

Limitations of existing detectors and the effectiveness of RAD under corpus evolution. As shown in Figure 4, in Loop 1 all retrieved texts are human-written; thus, detection errors in this iteration correspond purely to false positives, providing a clean assessment of the risk of mistakenly filtering human texts. Off-the-shelf detectors (OTS-Det) achieve only about 93% accuracy, indicating a non-negligible false positive rate (FPR=7%) that removes substantial amounts of relevant human texts. Such errors are particularly harmful in RAG systems, as discarded human texts cannot be recovered before generation, making OTS-Det unsuitable for deployment (Nicks et al., 2024; Wu et al., 2024). Directly fine-tuned detectors (FT-Det) substantially reduce false positives in early iterations, achieving an FPR of about 0.2%, comparable to RAD. However, their performance degrades in mid-to-late stages as retrieved texts increasingly mix human texts with prior LLM-generated texts, revealing the limited robustness of detectors trained on static RAG data. In contrast, detectors trained with RAD consistently maintain low FPR and high detection accuracy across iterations. As shown in Figure 5, RAD retains over 90% of human-written passages in late-stage retrieval results across both HC3 and MPU detector, whereas ***w/o*** D and FT-Det suffer from severe LLM text accumulation. Notably, the proportion of LLM texts in search results stabilizes after five iterations rather than continu-

ing to grow, demonstrating that RAD effectively mitigates the "Spiral of Silence" effect.

4.3 Mechanism Analysis

In this section, we investigate the underlying mechanisms of detection failure and robustness under iterative RAG, focusing on two research questions.

RQ1: Why does detection become increasingly difficult under iterative RAG? Inspired by prior observations on retrieval bias in evolving corpora (Dai et al., 2023), we analyze how the distribution of retrieved texts shifts across iterations using log-likelihood (LL), a widely used model-agnostic cue for LLM text detection (Bao et al., 2024; Wang et al., 2023). In the ***w/o D*** RAG setting, we compute the average token-level LL under a fixed language model M_θ (OPT-1.3B (Zhang et al., 2022)) as:

$$\mathrm{LL}(x) = \frac{1}{T}\sum_{t=1}^{T} \log P_{M_\theta}(x_t \mid x_{<t}). \quad (3)$$

As shown in Figure 6, early-stage LLM texts (e.g., Loop 2) exhibit a clear LL gap from human texts, making them relatively easy to detect, since they are solely generated from human texts and retain strong LLM-style distributional characteristics. However, as iterative RAG proceeds, retrieved LLM texts are increasingly generated under mixed contexts containing both human and prior LLM texts. Such mixed-source generation shifts the resulting generated texts away from the original LLM text distribution and toward that of human texts. As a result, LL-based distributional cues become less discriminative, explaining why detectors fail to generalize under corpus evolution, consistent with prior findings on iteratively paraphrased or style-shifted texts (Koike et al., 2024; Krishna et al., 2023). We observe the same trend when computing LL using a different language model such as Qwen3-8B, with consistent distributional convergence across iterations (Appendix E).

RQ2: How does RAD address this difficulty? We investigate how RAD mitigates detection failures under iterative RAG, focusing on how detector robustness depends on the round of synthetic data used for training. Figure 7 reports recall on human texts (in loop 1) and LLM texts from four generation modes across different loops. FT-Det is trained only on directly generated texts (DG), without exposure to generation conditioned on retrieved texts.

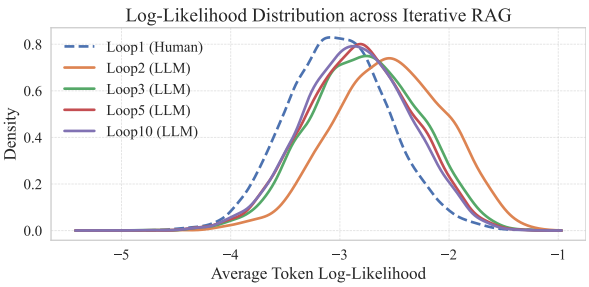

Figure 6: Kernel density estimation (KDE) of average token log-likelihood for retrieved texts across iterations.

As a result, their performance degrades when detecting RAG texts, indicating that training on direct query-based generation is insufficient to handle texts generated under iterative RAG.

We further compare RAD with two ablation variants. Specifically, RAD-1, RAD, and RAD-3 correspond to training detectors on LLM texts **synthesized at the first, second, and third rounds of corpus evolution**, respectively, as described in Section 3.3. **RAD-1**, which removes mixed-source synthesis, improves early-stage detection across all modes but still fails on later texts, indicating that generation diversity alone is insufficient. In contrast, our proposed **RAD** consistently improves recall on detecting LLM texts from later iterations while preserving high recall on human texts. **RAD-3**, which introduces additional synthesis rounds, further slightly increases recall on LLM texts but noticeably reduces recall on human texts, indicating over-alignment with human-like synthetic distributions. This trade-off is consistent with the LL shifting observed in Figure 6.

Overall, these results show that RAD is effective because it aligns detector training with mixed-source generation patterns characteristic of early corpus evolution, enabling robust performance in later iterations and reaches better trade-offs between detecting LLM and human texts. Beyond detection accuracy, we evaluate the system-level impact of different detector variants. As shown in Table 2, FT and RAD-1/3 fail to prevent long-term RAG degradation, whereas only RAD consistently stabilizes both retrieval and generation performance across iterations.

4.4 Further Analysis

We further evaluate the robustness of RAD under distribution shifts introduced by different components of the RAG pipeline, including retrievers, detector architectures, and text generators.

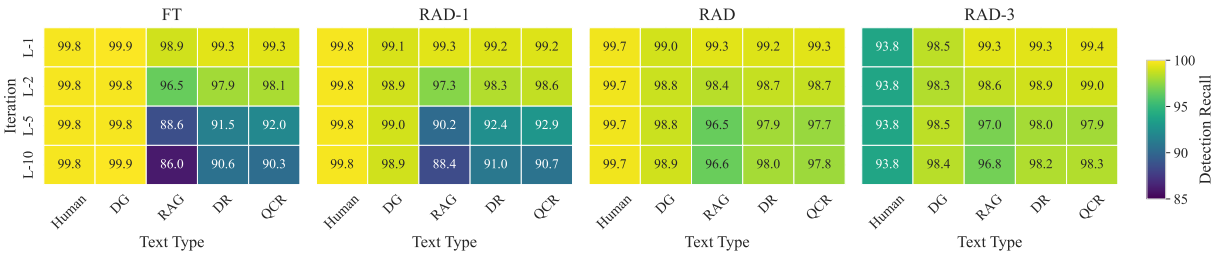

Figure 7: Recall on Human texts (in loop 1) and LLM texts across generation modes and corpus evolution.

Method	Acc@5	EM	EM_{llm}	Human@20
FT	64.8	61.1	41.9	73.7
RAD	**65.5**	**63.4**	**43.8**	**96.7**
RAD-1	65.0	61.5	42.8	81.3
RAD-3	65.1	61.7	42.6	84.4

Table 2: Impact of the synthesis stage used for detector training on RAG performance in Loop 10.

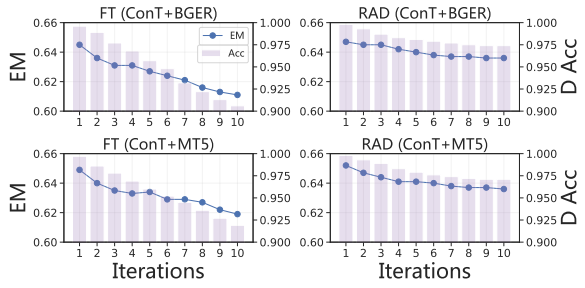

Figure 8: Different retrievers in the RAG pipeline with HC3 of FT and RAD paradigms.

Robustness across Retrievers We replace the retriever with Contriever and apply either BGE-Reranker or MonoT5 for re-ranking (denoted as Cont+BGER and Cont+MT5). As shown in Figure 8, FT-Det consistently degrade across retriever configurations, reflecting persistent source bias in retrieval pipelines (Dai et al., 2023). In contrast, RAD maintains stable detection performance over iterations under all retriever settings, demonstrating robustness to retrieval-induced distribution shifts.

Robustness across Detector Architectures. We further evaluate RAD on representative efficient detectors with different architectures and training paradigms, including MPU (Tian et al., 2024) and Fast-DetectGPT (Bao et al., 2024) (details in Appendix D). As shown in Table 5 in Appendix F, detector behavior varies across architectural design. Nevertheless, under the RAD paradigm, all detectors preserve over 90% human-written texts in the top-20 retrieval results at late stages, effectively mitigating the "Spiral of Silence" effect. HC3 exhibits the strongest overall alignment with the RAG setting, due to its QA-oriented formulation.

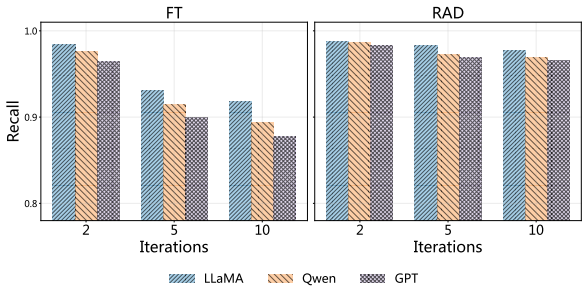

Figure 9: Detection recall on texts from different generators under FT and RAD paradigms.

Generalization across Generators. We train detectors on LLaMA-generated texts (training details are provided in Appendix D) and evaluate them on texts generated from LLaMA, Qwen, and GPT at loop 2, 5, and 10, simulating scenarios where the test-time generator is unknown. As shown in Figure 9, FT detectors exhibit progressively degraded recall over iterations, especially on GPT-4o-mini outputs, which are more difficult to distinguish. In contrast, detectors trained with RAD consistently maintain higher recall across generators and iterations, demonstrating strong cross-generator generalization in evolving RAG systems.

5 Conclusion

In this paper, we study the use of LLM text detectors in RAG systems to mitigate performance degradation and the "Spiral of Silence" effect caused by the accumulation of LLM texts. We propose RAD, a RAG-aware data augmentation strategy that synthesizes training data by mixing human and LLM texts under diverse RAG generation settings, aligning detector training with realistic corpus evolution. Experimental results and mechanism analysis show that RAD significantly improves detector robustness under iterative corpus evolution and effectively stabilizes both retrieval and generation

performance, highlighting the importance of data-centric detector training for reliable RAG systems.

Limitations

We acknowledge some limitations of our study:

First, while our work focuses on entirely excluding detected LLM-generated texts from the RAG generation context to prevent error amplification, future work may investigate how to handle these texts more effectively. Rather than simply discarding them all, hallucination detection methods (Min et al., 2023; Huang et al., 2026) could be employed to verify their factuality. In this way, the factually correct LLM-generated texts could be retained and presented to the user as optional or auxiliary references, without incorporating them into the RAG generation process. Second, while our study focuses on the widely adopted ODQA benchmarks for RAG systems, future work may explore the application of these detectors in other RAG scenarios or benchmarks.

Ethical Considerations

Our paper explores the impact of the LLM-generated texts on RAG-based systems and proposes solutions to mitigate the performance degradation and the "Spiral of Silence" phenomenon. We conduct our experiments on public datasets and do not release any generated texts during the iterative simulation process, thereby minimizing the risk of ethical issues.

Acknowledgment

This work was supported by National Natural Science Foundation of China No. 62272467 and the Beijing Major Science and Technology Project under Contract no. Z251100008425002. The work was partially done at the Beijing Key Laboratory of Research on Large Models and Intelligent Governance and Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

References

AI@Meta. 2024. Llama 3 model card.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1533–1544. ACL.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024a. Benchmarking large language models in retrieval-augmented generation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 17754–17762. AAAI Press.

Xiaoyang Chen, Ben He, Hongyu Lin, Xianpei Han, Tianshu Wang, Boxi Cao, Le Sun, and Yingfei Sun. 2024b. Spiral of silence: How is large language model killing information retrieval? - A case study on open domain question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14930–14951. Association for Computational Linguistics.

Xiaoyang Chen, Ben He, Hongyu Lin, Xianpei Han, Tianshu Wang, Boxi Cao, Le Sun, and Yingfei Sun. 2026. Breaking the spiral: A utility-driven optimization framework for balanced information retrieval in the llm era. *ACM Transactions on Information Systems*.

Zhihui Chen, Kai He, Yucheng Huang, Yunxiao Zhu, and Mengling Feng. 2025. DivScore: Zero-shot detection of LLM-generated text in specialized domains. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19242–19264, Suzhou, China. Association for Computational Linguistics.

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and unfairness in information retrieval systems: New challenges in the LLM era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 6437–6447. ACM.

Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, and Jun Xu. 2023. Llms may dominate information access: Neural retrievers are biased towards llm-generated texts. *CoRR*, abs/2310.20501.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,

Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *CoRR*, abs/2301.07597.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024, Salt Lake City, UT, USA, October 14-18, 2024*, pages 2251–2265. ACM.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: robust ai-text detection via adversarial learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. 2024. Are ai-generated text detectors robust to adversarial perturbations? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6005–6024. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232.

Yizheng Huang and Jimmy Huang. 2024. A survey on retrieval-augmented text generation for large language models. *CoRR*, abs/2404.10981.

Zhaoheng Huang, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. Enhancing LLM text detection with retrieved contexts and logits distribution consistency. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 9922–9934. Association for Computational Linguistics.

Zhaoheng Huang, Yutao Zhu, Jirong Wen, and Zhicheng Dou. 2026. Evaluating the factuality of large language models using multiple plug-and-play fact sources. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 41607–41609. AAAI Press.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 79 others. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1808–1822. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Minsang Kim, Cheoneum Park, and Seung Baek. 2024. Qpaug: Question and passage augmentation for open-domain question answering of llms. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 9024–9042. Association for Computational Linguistics.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, Canada.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Jesse G. Meyer, Ryan J. Urbanowicz, Patrick C. N. Martin, Karen O'Connor, Ruowang Li, Pei-Chen Peng, Tiffani J. Bright, Nicholas P. Tatonetti, Kyoung-Jae Won, Graciela Gonzalez-Hernandez, and Jason H. Moore. 2023. Chatgpt and large language models in academia: opportunities and challenges. *BioData Min.*, 16(1).

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Charlotte Nicks, Eric Mitchell, Rafael Rafailov, Archit Sharma, Christopher D. Manning, Chelsea Finn, and Stefano Ermon. 2024. Language model detectors are easily optimized against. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Elisabeth Noelle-Neumann. 1974. The spiral of silence a theory of public opinion. *Journal of communication*, 24(2):43–51.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Wang. 2023. On the risk of misinformation pollution with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1389–1403, Singapore. Association for Computational Linguistics.

Benji Peng, Keyu Chen, Ming Li, Pohsun Feng, Ziqian Bi, Junyu Liu, and Qian Niu. 2024. Securing large language models: Addressing bias, misinformation, and prompt attacks. *CoRR*, abs/2409.08087.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen, and Lei Meng. 2024. Rewritelm: An instruction-tuned large language model for text rewriting. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 18970–18980. AAAI Press.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. 2024. Multiscale positive-unlabeled detection of ai-generated texts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. Seqxgpt: Sentence-level ai-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1144–1156. Association for Computational Linguistics.

Yanting Wang, Wei Zou, Runpeng Geng, and Jinyuan Jia. 2025. Tracllm: A generic framework for attributing long context llms. In *USENIX Security Symposium*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2024. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *CoRR*, abs/2410.23746.

Han Xu, Jie Ren, Pengfei He, Shenglai Zeng, Yingqian Cui, Amy Liu, Hui Liu, and Jiliang Tang. 2024. On the generalization of training-based ChatGPT detection methods. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7223–7243, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Shiping Yang, Jie Wu, Wenbiao Ding, Ning Wu, Shining Liang, Ming Gong, Hengyuan Zhang, and Dongmei Zhang. 2025. Quantifying the robustness of retrieval-augmented language models against spurious features in grounding data. *Preprint*, arXiv:2503.05587.

Hongyeon Yu, Dongchan Kim, and Young-Bum Kim. 2026. Retrieval collapses when AI pollutes the web. In *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pages 8745–8748. ACM.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Yuqi Zhou, Sunhao Dai, Liang Pang, Gang Wang, Zhenhua Dong, Jun Xu, and Ji-Rong Wen. 2024. Source echo chamber: Exploring the escalation of source bias in user, data, and recommender system feedback loop. *CoRR*, abs/2405.17998.

Yutao Zhu, Zhaoheng Huang, Zhicheng Dou, and Ji-Rong Wen. 2025. One token can help! learning scalable and pluggable virtual tokens for retrieval-augmented large language models. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 26166–26174. AAAI Press.

1. Direct Generation (DG)
Provide a background document in 100 words according to your knowledge from Wikipedia to answer the given question.
Question: {Question}
Background Document:

2. Retrieval-Augmented Generation (RAG)
Context information is below.

{Retrieved_texts}

Using both the context information and also using your own knowledge, answer the following question with a background document in 100 words.
Question: {Question}
Background Document:

3. Direct Rewriting (DR)
Document is below.

{Retrieved_text}

Rewrite the document in 100 words.
Rewritten document:

4. Question-Centric Rewriting (QCR)
Document is below.

{Retrieved_text}

Rewrite the document in 100 words, preserving the original content and narrative details, while making it more relevant to the given question.
Question: {Question}
Rewritten document:

Figure 10: Prompts for LLM text generation methods.

A Prompts of Generation Methods

We provide the detailed prompts for four LLM text generation methods in Figure 10. We also provide cases of LLM-generated texts in Figure 11, showing that each generation method follow the instruction to generate texts.

B Post-Processing Details

Inspired by prior work (Chen et al., 2024b), we remove any specific phrases that indicate the texts are from LLMs through string matching. We collect over 40 sentences through manually annotation, and examples of phrases are follows:

- As an AI language model...
- I'd be happy to assist you...

Query: who does ben have a baby with in eastenders

Answers: [Lola Pearce]

Retrieved Texts:
[1] Ben Mitchell (EastEnders) to someone, telling them that Ben is all theirs. When Ben arrives…
[2] Honey Mitchell for those who have children with the condition. In 2006, EastEnders pledged that they…
[3] Ben Mitchell (EastEnders) Ben Mitchell (EastEnders) Ben Mitchell is a fictional character from the BBC…
[4] Billy Mitchell (EastEnders) a realistic way. The DSA have expressed their desire to…
[5] Billy Mitchell (EastEnders) could arise. “EastEnders” pledged to make every effort to portray…
…

1. Direct Generation (DG)
Background Document: EastEnders EastEnders is a long-running British soap opera created by Julia Smith and Tony Holland. The show premiered on February 19, 1985, and has been broadcast on BBC One ever since. Set in the fictional London Borough of Walford, the show revolves around the lives of the residents of Albert Square, a close-knit community with a rich history. The show is known for its in-depth storylines, complex characters, and gripping plot twists. Over the years, EastEnders has tackled various social issues, including racism, homophobia, and domestic violence, making it a staple

2. Retrieval-Augmented Generation (RAG)
Background Document: Ben Mitchell (EastEnders) is a fictional character from the BBC soap opera \"EastEnders\". The role has been played by five different actors, with Charlie Jones, Joshua Pascoe, and others portraying Ben at various stages of his life. In the context of the series, Ben has had romantic relationships with several characters, including Abi Branning and Paul Coker. Answer: Based on the provided context information and general knowledge of the EastEnders storyline, Ben Mitchell does not have a baby with anyone in the provided contexts. However, in the actual EastEnders storyline, Ben Mitchell has a son called Lex

3. Direct Rewriting (DR)
“Ben Mitchell (EastEnders)” Ben Mitchell is a fictional character from the BBC soap opera "EastEnders". He has been played by five different actors: Matthew Silver as an infant from 1996-1998, Morgan Whittle as a toddler from 1999-2001, Charlie Jones in 2006, Joshua Pascoe from 2010-2011, and took a break before rejoining."

4. Question-Centric Rewriting (QCR)
“Ben Mitchell (EastEnders)” Ben's relationships are strained due to his supposed affair with Paul, which Abi discovers. To save their relationship, Abi announces she is pregnant after being manipulated by Babe Smith. However, Ben refuses to have sex with her, indicating that he has no intention of having a child with Abi. This implies that Ben may not be the father of Abi's child. The document does not explicitly mention the identity of Ben's baby's mother, but it suggests a separation between Ben and Abi, leaving the possibility open for another character.

Figure 11: Cases of LLM-generated texts.

- According to my knowledge...
- As of my knowledge cutoff date...
- Based on the provided context information, ...

C Pseudo Code of the Pipeline

To better clarify the RAG framework with text detectors, we provide pseudo code in Algorithm 1.

D Implementation Details

Corpus Statistics. The initial corpus C^1 is the Wikipedia dump from December 20, 2018, containing approximately 21 million passages (Karpukhin et al., 2020). At each iteration, the detector processes $200 \times 4 \times 100 = 80$K re-ranked passages, with an average runtime of about 5 minutes.

Detectors. Detectors are either applied using their released checkpoints or fine-tuned from RoBERTa-base using the downstream training set or the proposed augmentation strategy, following their original training configurations. For the filtering stage, HC3 (Guo et al., 2023) and MPU (Tian et al., 2024) are fine-tuned for one epoch with a learning rate of 5×10^{-5} and a batch size of 32. We additionally consider a representative metric-based detector, Fast-DetectGPT (Bao et al., 2024), which does not require training data and outputs a detection score rather than probability. Specifically, it first measures the log likelihood of an input text, then applies token-level perturbations and computes the likelihood differences before and after perturbation. Following Bao et al. (2024), we calibrate the detection scores by estimating Gaussian distributions for human-written (D_0) and LLM-generated (D_1) texts with parameters (μ_0, σ_0) and (μ_1, σ_1) from labeled training data, and compute the LLM-generation probability for a detection

Algorithm 1 Simulation of RAG Framework with Text Detectors

Input: Detector D, initial corpus C^1, datasets Q, QA pair $(q, a) \in Q$, generators G, generation methods M, and number of iterations T

Output: Performance of Retrieval RP^t, Generation GP^t, and Detection Accuracy DA^t in each iteration $t \in T$

for $t \leftarrow 1$ to T **do**
  for $(q, a) \in Q$ **do**
    {Retrieval Phase}
    $P_q^t \leftarrow \text{Retrieve-Rerank}(q, C^t)$
    {Filtering Phase}
    $\tilde{P}_q^t \leftarrow \text{Filter}(D, P_q^t)$
    {Evaluation Phase}
    $RP^t \leftarrow \text{Eval}_{ret}(\tilde{P}_q^t, a)$
    $DA^t \leftarrow \text{Eval}_{det}(D, P_q^t)$
    $S_q^t \leftarrow \text{Form-Context}(\tilde{P}_q^t)$
    $r \leftarrow$ empty-list
    for $g \in G$ **do**
      $r \leftarrow r \cup \text{RAG}(g, q, S_q^t)$
    end for
    $GP^t \leftarrow \text{Eval}_{gen}(r, a)$
    {Generation Phase}
    $\tilde{C} \leftarrow$ empty list
    for $g \in G$ **do**
      $m \leftarrow \text{Random-Selection}(M)$
      $r \leftarrow \text{Post-Proc}(\text{GenRes}(g_m, q, S_q^t))$
      $\tilde{C} \leftarrow \tilde{C} + r$
    end for
    {Index Updating Phase}
    $C^{t+1} \leftarrow C^t \cup \tilde{C}$
  end for
end for=0

score x as:

$$p(D_1 \mid x) = \frac{\mathcal{N}(x; \mu_1, \sigma_1^2)}{\mathcal{N}(x; \mu_0, \sigma_0^2) + \mathcal{N}(x; \mu_1, \sigma_1^2)}. \quad (4)$$

For the FT-Det setting, we obtain $(\mu_0, \sigma_0) = (-0.545, 1.021)$ and $(\mu_1, \sigma_1) = (1.738, 1.005)$. For the RAD setting, the corresponding parameters are $(\mu_0, \sigma_0) = (-0.545, 1.021)$ and $(\mu_1, \sigma_1) = (0.762, 1.175)$. We use OPT-1.3B (Zhang et al., 2022) as the detector backbone to ensure low extra latency in the RAG system.

Training Data. To construct detector training data, we proportionally sample 30K queries from the training sets of NQ, WebQ, and TriviaQA (PopQA has no training set), resulting in 60K passages in total. This keeps the training scale comparable to the original HC3 dataset. We use LLaMA-3.1-8B-Instruct (AI@Meta, 2024) to generate LLM texts for training. Statistics of datasets is shown in Table 3.

Type	FT		RAD	
	# Total	Avg. Tokens	# Total	Avg. Tokens
Human	30,000	145.30	30,000	145.30
DG	30,000	122.80	7,500	122.32
RAG	0	N/A	7,500	124.56
QR	0	N/A	7,500	121.48
QCR	0	N/A	7,500	120.39

Table 3: Statistics of the training data.

Evaluation and Generation Phases. During evaluation, three LLMs are used across four datasets. As a result, each iteration adds $200 \times 4 \times 3 = 2.4$K newly generated LLM passages to the index, and produces an additional 2.4K RAG responses for evaluation. With $T = 10$ iterations, each experimental run generates a total of $(2.4\text{K} + 2.4\text{K}) \times 10 = 48$K LLM texts. All experiments are conducted on NVIDIA A800 GPUs. Each complete run takes approximately 8.6 hours for a given configuration, of which the detection phase accounts for about 0.83 hours ($\sim$10%).

Evaluation Metrics. We evaluate RAG systems from three perspectives: retrieval performance (Eval_{ret}), detection performance (Eval_{det}), and generation performance (Eval_{gen}), as defined in Algorithm 1. The retrieval performance $\text{Eval}_{ret}(\tilde{P}_q^t, a)$ (Acc@5) in dataset Q is defined as:

$$\text{Acc@5} = \frac{1}{|Q|} \sum_{(q,a) \in Q} I\left[a \in \bigcup_{i=1}^{5} \tilde{P}_{q,i}^t\right] \quad (5)$$

where $\tilde{P}_{q,i}^t$ is the i-th passages in the filtered results $\tilde{P}_q^t$, and a is the ground-truth answer.

The detection performance $\text{Eval}_{det}(D, P_q^t)$ is defined as:

$$\text{Eval}_{det}(D, P_q^t) = \frac{1}{|Q|} \frac{1}{|P_q^t|} \sum_{(q,a) \in Q} \sum_{s \in P_q^t} I[D(s) = f(s)], \quad (6)$$

where P_q^t is the top 100 unfiltered retrieved passages for query q, $f(s) \in \{0, 1\}$ is the ground-truth label (0: human, 1: LLM), and $D(s) = I[P^L(s) \geq P^H(s)]$ is the detector prediction.

For the generation performance $\text{Eval}_{gen}(D, P_q^t)$, the EM_{llm} (Chen et al., 2024b) and the EM metric are defined as:

$$\text{EM}_{llm}(r, a) = \begin{cases} 1, & \text{if } r \text{ contains and supports } a, \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{EM}(r, a) = I(a \in r). \tag{7}$$

Prompt: LLM-based Exact match

Does the following response support the answer to the question?
Question: {question_str}
Response: {response_str}
Answer: {answer_str}
Just answer 'yes' or 'no'.

The final score for a generation metric M (either EM or EM_{llm}) is defined as:

$$\text{Eval}_{gen} = \frac{1}{|Q|}\frac{1}{|G|}\sum_{(q,a)\in Q}\sum_{g\in G}\left(M[g(q, S_q^t), a]\right) \tag{8}$$

E Additional Analysis of Log-Likelihood

We further compute token-level log-likelihood using Qwen3-8B as the scoring model. As shown in Figure 12, we observe the same trend as with OPT-1.3B: the log-likelihood distributions of LLM-generated texts progressively shift toward those of human-written texts as RAG iterations proceed.

We additionally analyze the log-likelihood distributions of texts generated under different generation modes at Loop 10 in the *w/o D* setting, including DG, RAG, DR, QCR, measured by OPT-1.3B and Qwen3-8B. As shown in Figure 13 and Figure 15, DG exhibits the highest average log-likelihood gap from human texts, while RAG yields the lowest and closer to human texts. This indicates that, compared to other generation modes, RAG-generated texts exhibit the lowest log-likelihood under the scoring models, and are therefore more difficult to be distinguished from human texts. These findings further support our claim that mixed-source generation in RAG accelerates distributional coverage and fundamentally challenges detector generalization under iterative corpus evolution.

F Robustness across Detectors

We further evaluate the robustness of the proposed RAD strategy across several representative LLM text detectors, including MPU (Tian et al., 2024) and Fast-DetectGPT (Bao et al., 2024), all trained

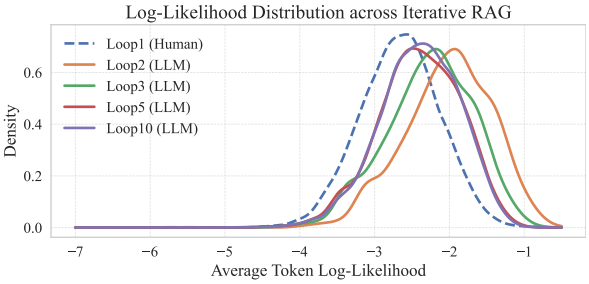

Figure 12: KDE of average token log-likelihood for retrieved texts across iterations, measured by Qwen3-8B.

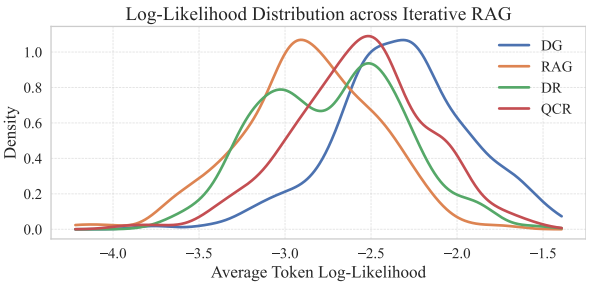

Figure 13: KDE of average token log-likelihood for generated texts, measured by OPT-1.3B.

under both the FT baseline and the RAD setting. MPU modifies the loss function to improve detection performance on short texts (typically fewer than 100 tokens). Fast-DetectGPT detects LLM texts by applying token-level perturbations and comparing the log-likelihood of the original text with that of its perturbed variants. As shown in Table 5, RAD yields the most significant gains when applied to the HC3 detector. One plausible explanation is that HC3 is specifically designed for QA scenarios, which are closely related to RAG settings. For the other detectors, RAD consistently improves performance over the FT baseline, although the improvements are comparatively more moderate.

G Analysis of Individual Generation Methods

To investigate how different generation methods affect the RAG system, we conduct experiments by exclusively using one specific generation method for all queries in the ***w/o*** D setting. The full experimental results are shown in Table 4 and Figure 14. We find that **all four generation methods detrimentally impact the RAG performance and gradually lead to the effect of "Spiral of Silence" over iterations.** Over time, as LLM texts increas-

Metric	Acc@5																
Dataset	NQ				WebQ				TQA				PopQA				Avg.
Loop	Ori.	L5	L10	Imp.	Ori.	L5	L10	Imp.	Ori.	L5	L10	Imp.	Ori.	L5	L10	Imp.	Imp.
Full	71.5	70.5	67.0	-6.3%	64.5	65.0	63.0	-2.3%	71.0	71.0	70.5	-0.7%	55.5	55.0	54.5	-1.8%	-2.8%
w/ DG	71.5	66.5	64.0	-10.5%	64.5	65.0	63.5	-1.6%	71.0	69.5	68.5	-3.5%	55.5	53.5	50.0	-9.9%	-6.4%
w/ RAG	71.5	71.0	65.5	-8.4%	64.5	63.5	61.0	-5.4%	71.0	71.5	71.0	+0.0%	55.5	55.5	54.0	-2.7%	-4.1%
w/ DR	71.5	70.0	69.5	-2.8%	64.5	63.5	64.0	-0.8%	71.0	70.5	70.5	-0.7%	55.5	54.5	54.5	-1.8%	-1.5%
w/ QCR	71.5	69.5	68.5	-4.2%	64.5	63.0	62.0	-3.9%	71.0	70.5	69.5	-2.1%	55.5	54.5	53.5	-3.6%	-3.5%
Metric	EM																
Full	63.8	59.8	58.3	-8.6%	60.8	58.8	57.0	-6.3%	78.0	75.7	73.7	-5.5%	55.0	54.8	52.7	-4.2%	-6.2%
w/ DG	63.8	58.2	56.4	-11.6%	60.8	57.9	55.7	-8.4%	78.0	74.3	71.9	-7.8%	55.0	52.8	49.6	-9.8%	-9.4%
w/ RAG	63.8	60.1	57.8	-9.4%	60.8	58.3	56.5	-7.1%	78.0	74.7	72.4	-7.2%	55.0	53.6	49.9	-9.3%	-8.3%
w/ DR	63.8	62.1	60.7	-4.9%	60.8	59.7	58.9	-3.1%	78.0	76.7	75.0	-3.8%	55.0	54.6	53.7	-2.4%	-3.6%
w/ QCR	63.8	58.6	57.6	-9.7%	60.8	58.4	57.1	-6.1%	78.0	75.0	73.5	-5.8%	55.0	53.1	52.3	-4.9%	-6.6%

Table 4: The impact of four LLM text generation methods on the RAG system. We evaluate the retrieval and generation performance during the corpus iteration process at the Loop 1 (Ori.), 5, and 10. and the improvement (Imp.) of Loop 10 compared to Loop 1.

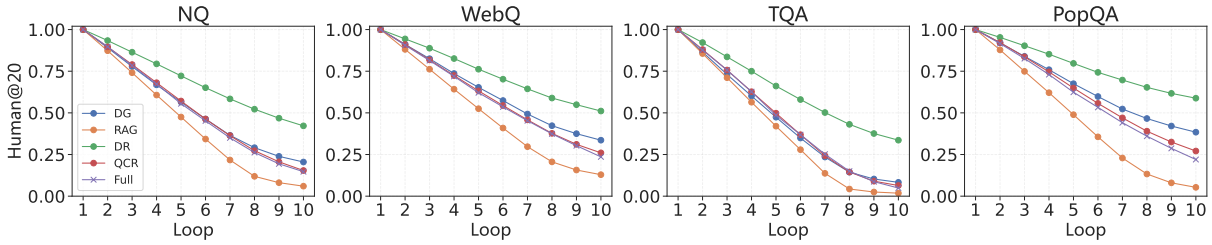

Figure 14: The percentage of human text in top 20 search results $\tilde{P}_q^t$ of the RAG system without the detector during corpus iteration when LLM-generated texts are gradually integrated into the corpus.

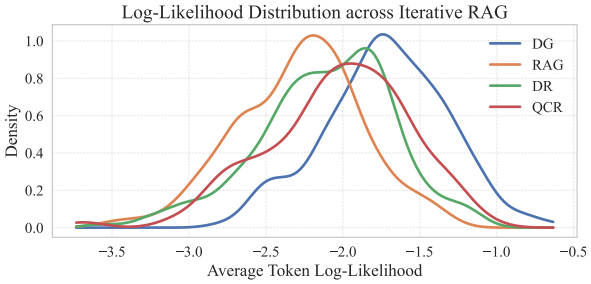

Figure 15: KDE of average token log-likelihood for generated texts, measured by Qwen3-8B.

Metric	Acc@5	EM	EM_{llm}	H@20
Loop 10 (late stage)				
w/o D	63.8	60.4	40.9	16.3
FT-HC3	64.8	61.1	41.9	73.7
FT-MPU	64.5	61.2	41.6	72.8
FT-FastDet	64.0	60.7	40.1	69.7
RAD-HC3	65.5	63.4	43.8	96.7
RAD-MPU	65.3	62.9	43.0	92.7
RAD-FastDet	65.0	63.0	43.3	93.5

Table 5: RAG performance with various detectors in the late stage (loop 10).

ingly infiltrate the search results, hallucinations dominate the input context, leading to a significant decline in performance across four datasets. This consistent degradation underscores the necessity of training text detectors on diverse data from all four generation methods, enabling detectors to filter out LLM texts in search results.

H Use of AI Assistants

We use ChatGPT to assist with writing clarity and presentation.[1]

[1] https://chatgpt.com/