LASTINGBENCH: Defend Benchmarks Against Knowledge Leakage

Yixiong Fang[♠*] **Tianran Sun**[♠*] **Yuling Shi**[♠] **Min Wang**[♡] **Xiaodong Gu**[♠†]

[♠] *Shanghai Jiao Tong University* [♡] *University of Pennsylvania*

{fangyixiong,seriousss,yuling.shi,xiaodong.gu}@sjtu.edu.cn

minyun@seas.upenn.edu

Abstract

The increasing complexity of large language models (LLMs) raises concerns about their ability to "cheat" on standard Question Answering (QA) benchmarks by memorizing task-specific data. This undermines the validity of benchmark evaluations, as they no longer reflect genuine model capabilities but instead the effects of data leakage. While prior work has focused on detecting such leakage, little attention has been given to mitigating its impact and preserving the long-term utility of benchmarks. In this paper, we introduce LASTINGBENCH, a novel framework designed to continuously reinforce and safeguard existing benchmarks against knowledge leakage. LASTINGBENCH identifies leakage points in the context through perturbation, then rewrites the leakage points to counterfactual ones—disrupting memorization while preserving the benchmark's original evaluative intent. Evaluations of state-of-the-art QA benchmarks show significant performance gaps, highlighting the efficacy of LASTINGBENCH in reducing memorization effects. LASTINGBENCH offers a practical and scalable solution to ensure benchmark robustness over time, promoting fairer and more interpretable evaluations of LLMs. Our code and data are available at https://github.com/Seriousss/LastingBench.

1 Introduction

The rapid advancement of LLMs has introduced critical challenges to the reliability and validity of QA evaluation benchmarks (Liu et al., 2024b; Wang et al., 2025a; Qian et al., 2025). Due to the opaque and large-scale nature of LLM training pipelines, these models often memorize parts of benchmark datasets (Xu et al., 2024c; Balloccu et al., 2024; Geva et al., 2023; Deng et al., 2023; Cheng et al., 2025). This unintended data leakage enables models to "cheat" during evaluation,

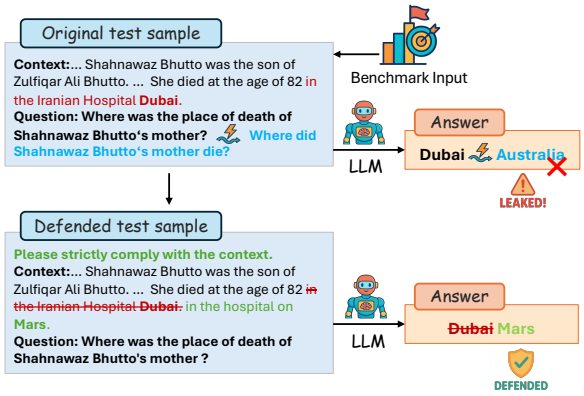

Figure 1: Overview of LASTINGBENCH.

producing correct answers without genuine understanding. As a result, evaluation scores on standard benchmarks may no longer reflect genuine model capabilities, but instead represent latent memorization effects. This poses a serious threat to fair and meaningful measurement in NLP research (Sainz et al., 2023a; Oren et al., 2023; Shi et al., 2023; Dekoninck et al., 2024b).

To address this issue, recent efforts have proposed various techniques to detect benchmark contamination, including perplexity-based analysis (Balloccu et al., 2024; Nasr et al., 2023), prompt-based probing (Deng et al., 2024; Oren et al., 2023), and guided completions (Golchin and Surdeanu, 2025). While useful for identifying leakage, these approaches do not offer long-term solutions, especially as LLMs continue to grow in size and training data volume.

A common reactive strategy is to retire leaked benchmarks and create new ones, as seen in systems like LiveCodeBench (Jain et al., 2024). However, this strategy is both costly and unsustainable, requiring continuous data collection and annotation (White et al., 2024; Rajore et al., 2024; Dong et al., 2024). Moreover, previously valuable benchmarks that are no longer maintained risk becoming

*Equal contribution.

†Correspondence: xiaodong.gu@sjtu.edu.cn

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 18304–18317

obsolete, leading to wasted effort and lost evaluative insights.

Therefore, the research community needs a robust pipeline not just for detecting contamination, but for actively recovering and reinforcing existing benchmarks (Musawi and Lu, 2025; Chen et al., 2025). Such a pipeline should enable benchmarks to remain reliable and resilient over time—regardless of future advances in model architectures, capabilities, or dataset scale (Dekoninck et al., 2024a; Jiang et al., 2024a). This would ensure evaluations focus on true improvements in reasoning and generalization, rather than incidental training data exposure.

In this paper, we propose LASTINGBENCH , a novel framework to **reinforce and safeguard existing benchmarks against knowledge leakage**, specifically targeting the long-context QA domain. Rather than continuously building new benchmarks, LASTINGBENCH enhances existing ones by identifying leakage points and rewriting them using counterfactuals, preserving the benchmark's original purpose while disrupting model memorization. Our methodology begins with perturbation-based detection. By systematically perturbing the original context and the question, we assess whether a model relies on internal memorization rather than contextual clues. When leakage is detected, we locate critical evidence segments—the minimal context required to justify the answer—using enriched chain-of-thought (CoT) queries from an LLM. These segments are then rewritten into plausible yet semantically contradictory counterfactuals. This makes the benchmark robust against memorization while maintaining its reasoning challenge.

We apply LASTINGBENCH to several long-context QA benchmarks and find widespread memorization, especially in HotpotQA. Our experiments reveal that larger models, such as GPT-4O and LLAMA-4-MAVERICK, exhibit higher levels of leakage. After applying our rewriting pipeline, we observe a substantial drop in inflated performance metrics, indicating a shift toward evaluations based on genuine reasoning and generalization.

Our contributions can be summarized as follows:

(i) We propose a solid leakage detection method, then reveal and empirically validate significant data leakage in long-context question-answering benchmarks.

(ii) We propose a chain-of-thought enhanced retrieval method using advanced reasoning models to pinpoint exact leakage locations within contexts.

(iii) We introduce a counter-fact paradigm to reconstruct benchmarks, preserving their structure and intent while minimizing vulnerability to model memorization.

(iv) We release modified benchmarks and demonstrate through extensive evaluations that, despite retaining original formats, models experience notable performance drops, confirming the effectiveness of our approach in evaluating genuine reasoning capabilities.

2 Knowledge Leakage: Detection and Current Landscape

In this section, we systematically investigate the phenomenon of knowledge leakage-whether a model's correct answers stem from memorized training data rather than contextual reasoning. Our study aims to identify and quantify large-scale knowledge leakage in existing long-context QA benchmarks.

2.1 Study Design

Given a long-context QA triple $(\mathbf{C}, q, a^\star)$—where $\mathbf{C} = \{c_1, \ldots, c_m\}$ is the concatenated context, q the original question, and $a^\star$ the gold answer—we study whether a language model $\mathcal{L}$ answers correctly by *memorization* rather than reasoning.

We propose two complementary techniques to identify knowledge leakage:

1) Context Perturbation: We exclude the context and prompt the model to answer the original question based solely on the question. If the model can still provide the correct answer, i.e., $\mathcal{L}(q) = a^\star$, this suggests its reliance on internal knowledge, indicating knowledge leakage.

2) Question Perturbation: We either rephrase the original question q into a semantically equivalent form $\tilde{q}$ or reformulate it into a logically contradictory version q_{con}, where the correct answer should conflict with the original. If the model answers the original question correctly but fails on the rephrased version, $\mathcal{L}(\tilde{q}, \mathbf{C}) \neq a^\star$, or provides the original answer even when the question is contradictory, $\mathcal{L}(q_{\text{con}}, \mathbf{C}) = a^\star$, this suggests dependence on memorized internal knowledge rather than contextual comprehension.

2.2 Experimental Setup

Dataset We conduct experiments on four QA datasets: 2WikiMQA (Ho et al., 2020), HotPotQA (Yang et al., 2018), Musique (Trivedi et al.,

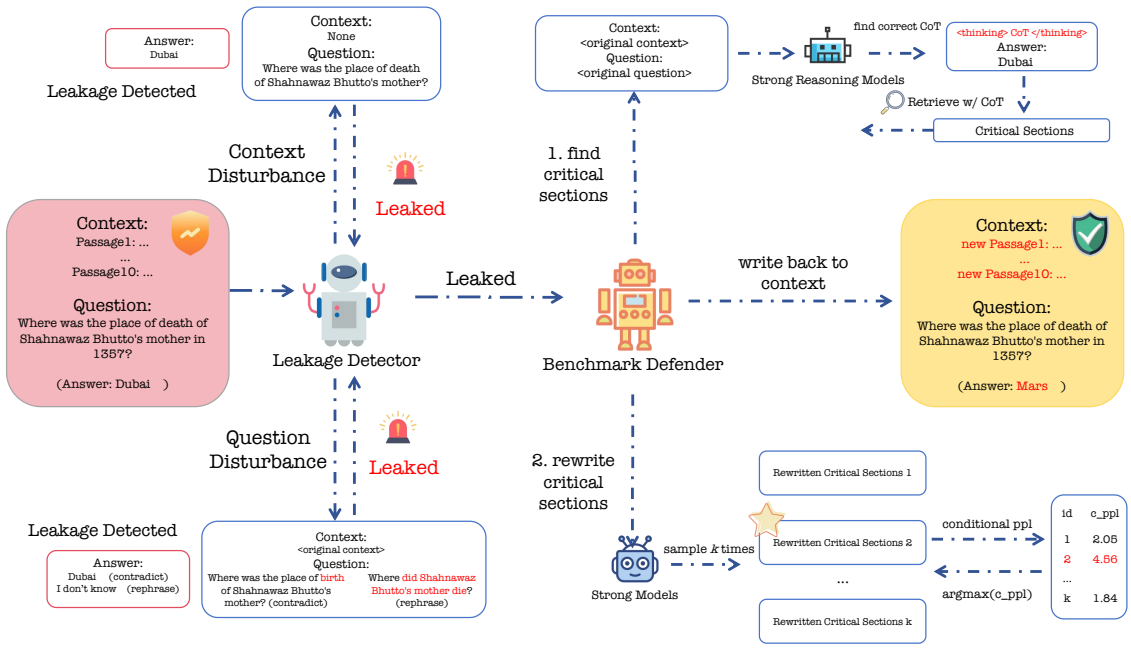

Figure 2: Illustration of LASTINGBENCH's pipeline: 1) Leakage Detection identifies memorization through context removal and question perturbation tests; 2) Critical Section Extraction uses strong reasoning models to locate essential evidence through CoT-enhanced retrieval; 3) Counterfactual Rewriting generates multiple alternative contexts and selects the optimal rewrite based on conditional perplexity, transforming factual information (e.g., "Dubai") into contradictory alternatives (e.g., "Mars") to effectively mitigate knowledge leakage while preserving the question's reasoning structure.

2022), and Multifieldqa_en (Bai et al., 2024a) from Longbench, including three multi-doc and one single-doc benchmarks. The average context length ranges from 5k to 60k. The diversity in context length, while ensuring sufficient coverage, enables a more comprehensive examination of data leakage issues in long-context QA.

Studied Models We select a diverse set of models with varying parameter sizes and architectures for our experiments.(1) Both open-source models (Qwen2.5-32B-Instruct (Qwen, 2025), Llama-3.1-8B-Instruct (et al, 2024), Phi-4 (Abdin et al., 2024)) and proprietary SOTA models including GPT-4o (Hurst et al., 2024) and DeepSeek-v3 (Liu et al., 2024a) are evaluated. (2) For the same model family, we also select models with different versions (e.g., Qwen3-32B vs. Qwen2.5-32B-Instruct) and parameter sizes(e.g., Qwen2.5-8B-Instruct vs. Qwen2.5-32B-Instruct), enabling multi-dimension comparisons.

Metrics We use two metrics to evaluate the accuracy of the model's answers following standard settings of existing benchmarks (Jiang et al., 2024a). **1) Exact Match (EM)** measures the percentage of predicted answers that exactly match the ground-truth answers. **2) F1 Score** measures the harmonic mean of precision and recall between the predicted and ground-truth answers, accounting for partial overlaps.

2.3 Results and Analysis

2.3.1 Results by Context Perturbation

As shown in Table 1, many models are able to answer a notable proportion of questions even without access to contextual information, indicating potential memorization. State-of-the-art LLMs, including GPT-4o and DEEPSEEK-V3, demonstrate surprisingly high accuracy on contextless queries—achieving scores of 0.35 on 2WikiMQA and 0.41 on HotpotQA. This strongly suggests that portions of these long-context datasets are memorized during pretraining. Interestingly, smaller open-source models such as QWEN3-8B and PHI-4 also achieve non-trivial accuracy under this setting, reinforcing the conclusion that dataset leakage is widespread and not limited to frontier models.

Across all evaluated datasets, some degree of leakage is observed. However, 2WikiMQA and HotpotQA show notably higher levels of leakage compared to MuSiQue. This discrepancy may be

Model	2WikiMQA	HotpotQA	MuSiQue
QWEN2.5-7B-INSTRUCT	0.24	0.26	0.09
QWEN2.5-32B-INSTRUCT	0.28	0.27	0.10
QWEN3-8B	0.27	0.28	0.07
QWEN3-32B	0.28	0.26	0.12
LLAMA-4-MAVERICK	0.30	0.37	0.13
GPT-4O	**0.35**	**0.41**	**0.27**
CLAUDE-3-HAIKU	0.23	0.37	0.14
PHI-4	**0.35**	0.27	0.12
DEEPSEEK-V3	0.33	0.39	0.17

Table 1: EM Scores for Context Removal Test Across Multiple Language Models and Datasets

Model	HotpotQA	Multifieldqa_en
QWEN2.5-7B-INSTRUCT	0.29	**0.20**
QWEN2.5-32B-INSTRUCT	0.37	0.11
QWEN3-8B	0.19	0.11
QWEN3-32B	0.21	0.07
GPT-4O	0.18	0.02
CLAUDE-3-HAIKU	**0.40**	0.19
DEEPSEEK-V3	0.13	0.07
PHI-4	0.34	0.14

Table 2: EM Scores When Evaluating Models with Contradictory Questions Against Original Ground Truth Answers

Model	2WikiMQA		HotpotQA	
	Original	Rephrased	Original	Rephrased
QWEN2.5-7B-INSTRUCT	0.49	0.52 (+0.03)	0.60	0.53 (–0.07)
QWEN2.5-32B-INSTRUCT	0.61	0.65 (+0.04)	0.65	0.62 (–0.03)
QWEN3-8B	0.83	0.50 **(–0.33)**	0.64	0.57 (–0.07)
QWEN3-32B	0.83	0.58 **(–0.25)**	0.64	0.52 (–0.12)
LLAMA-4-MAVERICK	0.67	0.67 (+0.00)	0.68	0.62 (–0.06)
GPT-4O	0.72	0.72 (+0.00)	0.69	0.57 (–0.12)
CLAUDE-3-HAIKU	0.60	0.70 (+0.10)	0.60	0.63 (+0.03)
DEEPSEEK-V3	0.72	0.67 (–0.05)	0.71	0.62 (–0.09)
PHI-4	0.63	0.80 (+0.17)	0.57	0.62 (+0.05)

Table 3: Performance Comparison Between Original Questions and Semantically Equivalent Reformulations Across Language Models

attributed to their earlier release dates, increasing the likelihood that they were included in the pre-training corpora of existing LLMs. Additionally, MuSiQue adopts a bottom-up strategy that systematically compose multi-hop questions from tightly coupled single-hop questions, resulting in tighter connection between the question and the context.

2.3.2 Results by Question Perturbation

Contradictory Question To test whether models truly comprehend the context or simply rely on memorized associations, we rephrase questions to convey meanings that contradict the original. The context remains unchanged, and a strong LLM[1] is used to generate these contradictory versions. Since some modified questions become unanswerable by design, we allow models to respond with *"I don't know"*. We then measure the proportion of cases where the model still outputs the original ground truth answer despite the contradiction.

The results shown in Table 2 reveal varying levels of memorization across models. Notably, CLAUDE-3-HAIKU shows the highest overlap on HotpotQA, with 40% of contradictory questions yielding the same answer as the original, despite the altered semantics. Similar trends are observed for models like QWEN2.5-32B-INSTRUCT and PHI-4. These findings suggest that current LLMs heavily rely on memorized internal answers rather than reasoning over the given context.

Equivalent Question We additionally evaluate model robustness to question rephrasing. The rephrased questions are semantically identical to the originals but differ in surface form. In principle, models with true comprehension should show minimal degradation in performance. As shown in Table 3, we observe notable performance drops on the rephrased questions. For instance, QWEN3-8B experiences a sharp decline of 0.33 EM on 2WikiMQA. Such a substantial drop suggests that the model's original performance may have relied more on memorization than on understanding the interplay between context and question. These results further highlight the fragility of benchmark performance under even minor input variations, revealing potential overfitting to known data.

[1]We use GPT-4O for rephrasing.

2.4 Current Landscape and Trend

Our findings reveal that knowledge leakage is already widespread across long-context benchmarks, and the situation is deteriorating. Newer and larger models such as Qwen3 exhibit much higher leakage than their predecessors, as evidenced by drastic performance drops under question rephrasing. This trend suggests that mainstream benchmarks gradually no longer reliably reflect models' genuine abilities—instead, they often measure memorization and result in inflated numbers. These findings highlight the urgent need for a pipeline to restore and defend benchmarks.

3 Method

To tackle the memorization problem of existing benchmarks, we propose a novel framework to continuously defend existing benchmarks against

Algorithm 1 LASTINGBENCH: Detecting Knowledge Leakage (DETECT) and Defending Benchmarks (DEFENSE)

function DETECT(q, $a^\star$, C, $\mathcal{L}$)
  $\tilde{q} \leftarrow$ REPHRASE(q) ▷ rephrase query
  $q_{\text{con}} \leftarrow$ CONTRA(q) ▷ contradictory query
  if $\mathcal{L}(q) = a^\star$ or $L(\tilde{q}, C) \neq a^\star$ or $L(q_{\text{con}}, C) = a^\star$ **then**
    return true
  end if
  return false
end function
function DEFENSE(q, $a^\star$, C, $\mathcal{L}$, k)
  $\tilde{q} \leftarrow$ REPHRASE(q)
  repeat ▷ find critical context
    $(\tilde{a}, r) \leftarrow L(\tilde{q}, C)$
    $q^+ \leftarrow (\tilde{q}, \tilde{a}, r)$
    $C_{\text{crit}} \leftarrow$ RETRIEVE(C, q^+)
  until $L(\tilde{q} \mid C_{\text{crit}}) = a^\star$
  for $i \leftarrow 1$ **to** k **do** ▷ counterfactual rewriting
    $C_{\text{cf}}^{(i)} \leftarrow$ REWRITECONTRADICT(C_{crit})
    $\text{CPPL}_i \leftarrow \text{PPL}(q) - \text{PPL}(q \mid C_{\text{cf}}^{(i)})$
  end for
  $i^\star \leftarrow \arg\max_i \text{CPPL}_i$
  return MERGE(C, C_{crit}, $C_{\text{cf}}^{(i^\star)}$)
end function

knowledge leakage. Given a QA instance $(\mathbf{C}, q, a)$ identified as potentially leaking knowledge (as described in Section 2.1), LASTINGBENCH applies a two-stage defense process: (1) Localize critical evidence segments within the context $\mathbf{C}$ that are likely memorized; and (2) Replace these segments with carefully constructed counterfactuals that contradict model-internal knowledge, while preserving the question's evaluative intent. An overview of the pseudocode is presented in Algorithm 1.

3.1 Localize Critical Sections

After detecting potential knowledge leakage, we localize critical section within the original context. Given a QA instance with context $\mathbf{C}$, question q, and answer a, we aim to identify a minimal context segment $\mathbf{C}_{\text{crit}} \subseteq \mathbf{C}$ that suffices to answer q (or a paraphrased equivalent). To reduce the risk of triggering memorized responses, we reformulate the question into a semantically equivalent version, denoted $\tilde{q}$.

We start by prompting a strong reasoning model $\mathcal{L}$ (e.g., DEEPSEEK-R1 (DeepSeek-AI, 2025)) to answer $\tilde{q}$ using Chain-of-Thought (CoT) reasoning (Jiang et al., 2025), producing an intermediate answer $\tilde{a} = \mathcal{L}(\tilde{q}, \mathbf{C})$ and a reasoning trace $\mathbf{r} = \text{CoT}(\mathcal{L}, \tilde{q}, \mathbf{C})$. The correctness of $\tilde{a}$ indicates that the reasoning path has successfully captured key semantic cues. Next, We construct an enriched retrieval query by concatenating $\mathbf{r}$, $\tilde{q}$ and $\tilde{a}$, denoted q^+.

$$q^+ = \text{CONCAT}(\mathbf{r}, \tilde{q}, \tilde{a}) \tag{1}$$

Using the enriched query q^+, we perform *embedding-based retrieval*: each chunk $c_j \in \mathbf{C}$ and the query itself are embedded by the sentence-encoder $f(\cdot)$, cosine similarities are computed, and the top-k most similar chunks are retained. The resulting minimal evidence set is

$$\begin{aligned} \mathbf{C}_{\text{crit}} &= \bigcup_{j \in \mathcal{N}_k} c_j, \\ \mathcal{N}_k &= \underset{\text{sim}}{\text{Top-k}}\big(f(q^+), \{f(c_1), \ldots, f(c_m)\}\big). \end{aligned} \tag{2}$$

To validate that $\mathbf{C}_{\text{crit}}$ indeed contains sufficient information, we use $\mathcal{L}$ to answer $\tilde{q}$ using only $\mathbf{C}_{\text{crit}}$. If the model fails to produce the correct answer, meaning the retrieval omitted crucial content, we repeat the process with an updated retrieval query until a satisfactory $\mathbf{C}_{\text{crit}}$ is found.

3.2 Counterfactual Rewriting

To prevent reliance on memorized context segments, we apply a counterfactual rewriting strategy that transforms the localized critical evidence into content that contradicts the model's internal knowledge. This serves two purposes: (1) to weaken the model's ability to rely on memorized knowledge, and (2) to test whether the model can adapt to the revised context while preserving the original question's reasoning requirements.

For each critical section $\mathbf{C}_{\text{crit}}$, we generate k counterfactual variants $\{\mathbf{C}_{\text{cf}}^{(i)}\}_{i=1}^{k}$ using a predefined rewriting paradigm[2]. Each $\mathbf{C}_{cf}^{(i)}$ is a contexually consistent but semantically conflicting alternative to $\mathbf{C}_{crit}$.

To select the most effective counterfactual, we compute the conditional perplexity of the original question q as:

$$\text{CPPL}(\mathbf{C}_{\text{cf}}^{(i)}, q) = \text{PPL}(q) - \text{PPL}(q \mid \mathbf{C}_{cf}^{(i)})$$

[2]Details of prompts are provided in Appendix B

where $\text{PPL}(q)$ is the perplexity of the question q answered without context, and $\text{PPL}(q \mid \mathbf{C}_{cf}^{(i)})$ measures perplexity given the i-th counterfactual. A higher CPPL indicates a stronger contradiction to the model's internal knowledge. We select the counterfactual with the highest conditional perplexity:

$$\mathbf{C}_{\text{cf}}^{*} = \arg\max_{i} \text{CPPL}(\mathbf{C}_{\text{cf}}^{(i)}, q)$$

Finally, we construct the defended context $\mathbf{C}^{\text{defend}}$ by replacing $\mathbf{C}_{\text{crit}}$ in $\mathbf{C}$ with $\mathbf{C}_{\text{cf}}^{*}$. This defended instance enforces contextual alignment and tests the model's ability to reason rather than recall, preserving benchmark reliability.

4 Experiments

In this section, we evaluate LASTINGBENCH through experiments. We use the same experimental setup as in Section 2.

4.1 Revised Dataset Details

We utilize DEEPSEEK-R1 as the leakage detecting model, evaluating with same methods in Section 2. Table 4 reports the percentages of revised (leaked) and unchanged entries for each dataset used in our experiments. Regarding the quality of the rewritten dataset, through manual inspection, we found that after rewriting, 99% of the contexts can fully support answering the question with the antifact answer. A genuine understanding of the full context should enable the model to generate the antifact answer. We acknowledge that a very small portion contained minor issues, such as unnecessary alterations to irrelevant parts. However, in most cases, the inaccurate modifications only involve distracting content that does not affect answering the question.

Dataset	Revised Entries	Unchanged Entries
2WIKIMQA	81%	19%
HOTPOTQA	71%	29%
MUSIQUE	61%	39%
MULTIFIELDQA_EN	77%	23%

Table 4: Distribution of Revised and Preserved Entries Across Evaluated Datasets

4.2 Efficacy in Model Evaluation

We show the efficacy of LASTINGBENCH in model evaluation by comparing model performance on the original and the revised benchmarks.

As is displayed in Table 5, across all datasets, most models exhibit a consistent performance decline across different datasets. The performance drop is more pronounced on HotpotQA and 2WikiMQA, with HotpotQA showing the most significant decline (**30%** for Deepseek-v3). This aligns with our observations in §2.3.1 regarding context perturbation-based leakage detection. Among all evaluated models, Claude-3-Haiku (**27%** on HotpotQA) and the Qwen 3 series (**30%** on 2WikiMQA) show more pronounced performance degradation on the revised datasets, which is consistent with the leakage detection results presented in §2.3.2. This may be attributed to the fact that these models are relatively newer and therefore more likely to have been exposed to these datasets during the training stage, leading to potentially inflated performance. In addition, larger models tend to exhibit a higher degree of data memorization within their parameters. For example, GPT-4o and DeepSeek-v3 show a consistent decline on all the datasets. In contrast, our revised datasets provide a more trustworthy assessment of the models' genuine long-context reasoning ability.

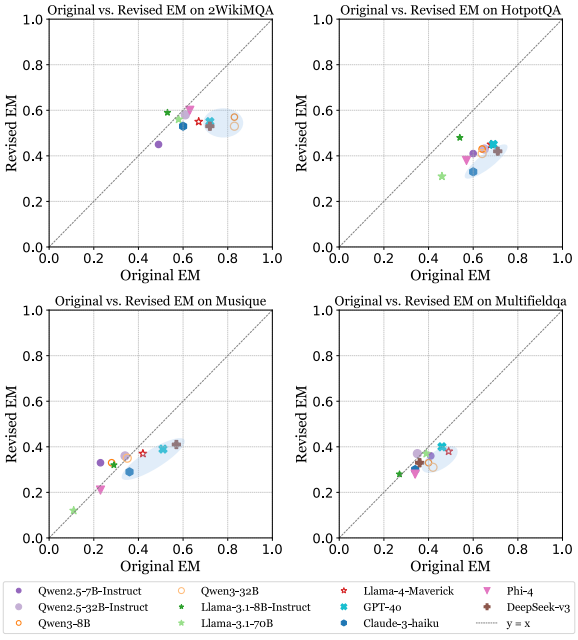

Figure 3: Performance Comparison of Language Models on Original and LastingBench-Revised Datasets

4.3 Comparative Analysis

We then compare LASTINGBENCH to alternative methods and analyze the efficacy of LASTINGBENCH. More specifically, we substitute the counterfactual rewriting component with a random

Model	2WikiMQA				HotpotQA				Musique				Multifieldqa_en			
	Original		Defensed		Original		Defensed		Original		Defensed		Original		Defensed	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
QWEN2.5-7B-INSTRUCT	0.48	0.49	0.45	0.45	0.56	0.60	0.41	0.41	0.26	0.23	0.33	0.33	0.53	0.41	0.42	0.36
QWEN2.5-32B-INSTRUCT	0.61	0.61	0.58	0.58	0.61	0.65	0.43	0.43	0.36	0.34	0.36	0.36	0.47	0.35	0.46	0.37
QWEN3-8B	0.76	0.83	0.56	0.57	0.60	0.64	0.45	0.43	0.31	0.28	0.33	0.33	0.49	0.40	0.46	0.33
QWEN3-32B	0.76	0.83	0.53	0.53	0.56	0.64	0.40	0.41	0.35	0.35	0.36	0.35	0.51	0.42	0.43	0.31
LLAMA-3.1-8B-INSTRUCT	0.28	0.53	0.39	0.59	0.32	0.54	0.26	0.48	0.16	0.29	0.13	0.32	0.38	0.27	0.30	0.28
LLAMA-3.1-70B	0.46	0.58	0.46	0.56	0.36	0.46	0.28	0.31	0.08	0.11	0.12	0.12	0.50	0.39	0.47	0.37
LLAMA-4-MAVERICK	0.63	0.67	0.52	0.55	0.64	0.68	0.44	0.45	0.44	0.42	0.38	0.37	0.54	0.49	0.49	0.38
GPT-4O	0.68	0.72	0.55	0.55	0.67	0.69	0.47	0.45	0.48	0.51	0.39	0.39	0.56	0.46	0.53	0.40
CLAUDE-3-HAIKU	0.36	0.60	0.40	0.53	0.47	0.60	0.28	0.33	0.22	0.36	0.18	0.29	0.40	0.34	0.41	0.30
PHI-4	0.23	0.63	0.23	0.60	0.33	0.57	0.14	0.38	0.08	0.23	0.06	0.21	0.44	0.34	0.36	0.28
DEEPSEEK-V3	0.64	0.72	0.50	0.53	0.68	0.72	0.44	0.42	0.53	0.57	0.41	0.41	0.50	0.36	0.46	0.33

Table 5: Comprehensive Performance Analysis of Models on Original and LASTINGBENCH-Defensed Datasets with Change Visualization (Red Indicates Performance Decrease, Green Indicates Increase)

Model	Original	Random	Counterfact
QWEN2.5-7B-INSTRUCT	0.60	0.41	0.41
QWEN2.5-32B-INSTRUCT	0.65	0.50	0.43
QWEN3-8B	0.64	0.44	0.43
QWEN3-32B	0.64	0.47	0.41
LLAMA-3.1-8B-INSTRUCT	0.54	0.42	0.48
LLAMA-3.1-70B	0.46	0.44	0.31
GPT-4O	0.69	0.49	0.45
CLAUDE-3-HAIKU	0.60	0.43	0.33
PHI-4	0.50	0.44	0.38
DEEPSEEK-V3	0.71	0.50	0.42

Table 6: Comparative Analysis of Model Performance (EM) Across Different Context-Rewriting Strategies

Figure 4: Training Loss Comparison of Language Models on Original and LASTINGBENCH-Revised 2WikiMQA Datasets

rewriting strategy: prompting the model to generate a random alternative answer and accordingly rewrite the supporting evidence, while the remainder of the pipeline remains unchanged.

Table 6 shows the model performance on HotpotQA. The results indicate that model performance on the randomly reformulated dataset falls short of the original, yet surpasses that on the counterfactual dataset, which suggests that our counterfactual rewriting method is more resistant to model cheating and poses a greater challenge, making it less likely to be memorized during pre-training. As a result, it offers a more faithful and enduring evaluation of the model's long-context reasoning abilities.

4.4 Efficacy in Model Training

To examine whether our dataset defense effectively increases the learning complexity for language models and consequently reduces the likelihood of models memorizing information during the training stage, we conducted an empirical evaluation. Specifically, we fine-tuned four distinct models on both the original 2WikiMQA dataset and our revised dataset[3]. Introducing knowledge conflicts in datasets naturally creates greater complexity during the learning process. Ideally, this discourages models from memorizing knowledge.

As displayed in Figure 4, throughout the fine-tuning process, all models exhibit higher training loss on our revised dataset than on the original dataset, indicating that it exposes them to knowledge conflict situations. The consistently higher training loss values indicate that the revised data poses greater learning difficulty, which in turn mitigates the risk of future data leakage by making it less likely for models to memorize or retain such

[3]Details of our fine-tuning settings are provided in Section A

information.

4.5 Case Study

Table 7 illustrates how models may produce correct answers without genuinely reasoning over the provided context. In the first setting (Without Context), the model still predicts the correct year of appointment and the correct Olympic Games even though no supporting passages are given. This suggests that the model simply recalled memorized knowledge rather than performing contextual reasoning.

In contrast, the second part (Original/Rewritten Context) highlights the effect of counterfactual rewriting. When the original context explicitly contains the correct evidence (**Marie of Hohenstaufen**), the model outputs the right answer. However, after rewriting the critical evidence to state **Joan of Arc**, the model still produces the memorized answer, thus failing on the modified case. This discrepancy reveals that the earlier success was driven by leakage and memorization rather than true understanding of the context.

5 Related Works

5.1 Contamination Detection

The evaluation of Large Language Models (LLMs) is increasingly complicated by data contamination—the unintended overlap between training data and evaluation benchmarks (Fu et al., 2025; Cheng et al., 2025; Xu et al., 2024a). This issue can artificially inflate performance scores, misrepresenting LLMs' true generalization capabilities. Vast pre-training datasets, often web-scraped, frequently include common benchmark samples, leading to deceptively high leaderboard scores that may not reflect real-world robustness (Tonmoy et al., 2024; Xu et al., 2024b; Fu et al., 2025).

Early detection efforts relied on simple string-matching and overlap detection to find direct textual similarities. However, these methods struggle with nuanced contamination, like paraphrased content or memorized knowledge lacking direct textual overlap (Carlini et al., 2021; Nasr et al., 2023). Consequently, researchers developed advanced techniques that do not require access to proprietary training data (Musawi and Lu, 2025; Dekoninck et al., 2024b; Deng et al., 2023). For instance, PaCoST (Paired Confidence Significance Testing) constructs distributionally similar counterparts for benchmark instances and statistically analyzes model confidence on original versus counterpart data to detect significantly higher confidence on originals as potential contamination (Zhang et al., 2024; Li et al., 2023). Similarly, ConStat adopts a performance-based approach, defining contamination as artificially inflated, non-generalizable performance, and uses statistical analysis to compare model performance on primary and reference benchmarks (Dekoninck et al., 2024a; Bommasani et al., 2021).

Despite these advancements, comprehensively addressing data contamination remains challenging. Notably, specific methodologies for contamination detection in long-context scenarios are lacking in reviewed material (Wang et al., 2024; Jiang et al., 2024a; Geva et al., 2023). The complexities of extended input sequences may demand novel approaches for identification and mitigation in such contexts (Rajore et al., 2024; Shi et al., 2023).

5.2 Long Context Benchmark

Long-context benchmark suites now serve as the standard yardstick for a broad range of evaluations—spanning RAG applications (Cheng et al., 2024; Fang et al., 2025; Wang et al., 2025c), context-compression methods (Jiang et al., 2023; Pan et al., 2024; Jiang et al., 2024b), and many other long-sequence tasks (Wang et al., 2025b; Zhu et al., 2025; Liu et al., 2025b; Lan et al., 2025; Han et al., 2024; Liu et al., 2025a; Yang et al., 2025; Qiu et al., 2025; Zhang et al., 2025b,a). The development of long-context benchmarks like SCROLLS (Shaham et al., 2022) and LongBench (Bai et al., 2024b) has been crucial for evaluating LLMs' extended sequence processing. However, these suites are also susceptible to data contamination, where inadvertent inclusion of evaluation samples in training data can inflate performance metrics (Xu et al., 2024d; Oren et al., 2023).

To address this, newer contamination-aware benchmarks have been developed, employing various strategies to combat data leakage. Some, like BAMBOO (Dong et al., 2024), LiveBench (White et al., 2024), and AcademicEval (Chen et al., 2025), feature continuously updated test sets. Others, such as VarBench, introduce dynamic variable perturbation to create more robust and generalizable evaluation scenarios (Qian et al., 2024). A significant limitation with manually updated benchmarks, however, is their heavy reliance on laborious human effort for data collection and maintenance, rendering the process resource-intensive (Sainz et al., 2023b).

Without Context:	After Russell D. Moore served at the Southern Baptist Theological Seminary, he became the President of The Ethics & Religious Liberty Commission (ERLC) in what year?	**GPT-4o: 2013** ✓
	Professional cyclist Sara Symington competed in which Olympic Games held in Sydney, Australia?	**GPT-4o: 2000** ✓
Original/rewritten Context:	**Question**: Who is the paternal grandmother of Marie of Brabant, Queen of France?	
	Original Context: ... Marie of Brabant (13 May 1254 – 12 January 1322) was Queen of France from 1274 until 1285 as the second wife of King Philip III. Born in Leuven, Brabant, she was a daughter of Henry III, Duke of Brabant, and Adelaide of Burgundy. ... Henry III of Brabant (c. 1230 – February 28, 1261, Leuven) was Duke of Brabant between 1248 and his death. **He was the son of Henry II of Brabant and Marie of Hohenstaufen.** ...	**Marie of Hohenstaufen**
	Rewritten Context: ... Marie of Brabant (13 May 1254 – 12 January 1322) was Queen of France from 1274 until 1285 as the second wife of King Philip III. Born in Leuven, Brabant, she was a daughter of Henry III, Duke of Brabant, and Adelaide of Burgundy. ... Henry III of Brabant (c. 1230 – February 28, 1261, Leuven) was Duke of Brabant between 1248 and his death. **He was the son of Henry II of ~~Brabant and Marie of Hohenstaufen.~~ Joan of Arc** ...	~~**Marie of Hohenstaufen**~~ **Joan of Arc**

Table 7: Case study showing knowledge leakage and counterfactual defense. Top: without any context, the model still gives correct answers (2013, 2000), revealing reliance on memorized knowledge. Bottom: in the **Original Context**, the correct answer (**Marie of Hohenstaufen**) is present, so the model answers correctly (✓), but this could be due to memorization rather than reasoning. In the **Rewritten Context**, we replace the key evidence with Joan of Arc. The model still outputs the original memorized answer, yielding an error (×), which confirms that the first success came from recall instead of contextual understanding.

To mitigate this, approaches like Antileak-Bench aim to automatically construct benchmarks from updated real-world knowledge, offering a more scalable solution (Wu et al., 2024). Ensuring these updated benchmarks remain truly contamination-free is also challenging, as new data might still contain pre-existing knowledge (Yuan et al., 2024; Lee et al., 2024; Gao et al., 2025). Furthermore, undisclosed training data specifics for many LLMs make guaranteeing contamination absence in these benchmarks difficult (Modarressi et al., 2025; Qi et al., 2024).

6 Conclusion

In this paper, we introduced LASTINGBENCH, a novel framework designed to address the critical challenge of model cheating and knowledge leakage in QA benchmarks. LASTINGBENCH combines perturbation-based detection with counterfactual rewriting to identify and repair leakage points—disrupting memorization while preserving the original evaluative intent of the benchmark. Our experiments on long-context QA benchmarks reveal widespread data leakage across a range of models, including both frontier and smaller open-source LLMs. By applying LASTINGBENCH, we substantially reduce memorization effects and provide a more faithful evaluation of models' reasoning and generalization capabilities. This work offers a robust and sustainable approach to reinforce benchmark integrity, ensuring that evaluations remain meaningful and resilient against the evolving capabilities of LLMs.

Limitations

LASTINGBENCH currently targets with-context QA and has only been validated on textual long-context benchmarks; applying the same leakage probes and counterfactual rewriting to generation-heavy or multimodal tasks (e.g., summarization, dialogue, code, images) will require new success criteria and may introduce extra computational overhead. Addressing these extensions is left for future work.

Acknowledgment

This research is funded by the National Key Research and Development Program of China (Grant No. 2023YFB4503802) and the Natural Science Foundation of Shanghai (Grant No. 25ZR1401175).

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero

Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 technical report. Preprint, arXiv:2412.08905.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024a. LongBench: A bilingual, multitask benchmark for long context understanding. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. Preprint, arXiv:2412.15204.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

Nicholas Carlini, Florian Tramer, Eric Wallace, Miroslav Jagielski, Matthew Roberts, Samuel R Bowman, Moritz Hardt, Nicolas Papernot, et al. 2021. Extracting training data from large language models. In 30th USENIX Security Symposium (USENIX Security 21), pages 263–280.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and Baishakhi Ray. 2025. Recent Advances in Large Langauge Model Benchmarks against Data Contamination: From Static to Dynamic Evaluation. Preprint, arXiv:2502.17521.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. Preprint, arXiv:2405.13792.

Yuxing Cheng, Yi Chang, and Yuan Wu. 2025. A Survey on Data Contamination for Large Language Models. Preprint, arXiv:2502.14425.

Michael Han Daniel Han and Unsloth team. 2023. Unsloth.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Preprint, arXiv:2501.12948.

Jasper Dekoninck, Mark Niklas Mueller, and Martin Vechev. 2024a. ConStat: Performance-Based Contamination Detection in Large Language Models. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.

Jasper Dekoninck, Mark Niklas Müller, Maximilian Baader, Marc Fischer, and Martin Vechev. 2024b. Evading data contamination detection for language models is (too) easy. arXiv preprint arXiv:2402.02823.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2023. Investigating data contamination in modern benchmarks for large language models. arXiv preprint arXiv:2311.09783.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Benchmark probing: Investigating data leakage in large language models. In NeurIPS 2023 Workshop on Backdoors in Deep Learning - The Good, the Bad, and the Ugly.

Zican Dong, Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2024. BAMBOO: A Comprehensive Benchmark for Evaluating Long Text Modeling Capacities of Large Language Models. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 2086–2099, Torino, Italia. ELRA and ICCL.

Aaron Grattafiori et al. 2024. The llama 3 herd of models. Preprint, arXiv:2407.21783.

Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. 2025. Attentionrag: Attention-guided context pruning in retrieval-augmented generation. Preprint, arXiv:2503.10720.

Yujuan Fu, Ozlem Uzuner, Meliha Yetisgen, and Fei Xia. 2025. Does Data Contamination Detection Work (Well) for LLMs? A Survey and Evaluation on Detection Assumptions. In Findings of the Association for Computational Linguistics: NAACL 2025, pages 5235–5256, Albuquerque, New Mexico. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Wenlong Wu, Zijing Huang, Bohan Li, and Haofen Wang. 2025. U-niah: Unified rag and llm evaluation for long context needle-in-a-haystack. arXiv preprint arXiv:2503.00353.

Mor Geva, Tal Schuster, Jonathan Berant, and Omer Levy. 2023. Detecting pretraining data from large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1207–1220. Association for Computational Linguistics.

Shahriar Golchin and Mihai Surdeanu. 2025. Data contamination quiz: A tool to detect and estimate contamination in large language models. Preprint, arXiv:2311.06233.

Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Donglin Wang, Honggang Chen, Qingsen Yan, and Siteng Huang. 2024. Filter, correlate, compress: Training-free token reduction for mllm acceleration. arXiv preprint arXiv:2411.17686.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Proceedings of the 28th International Conference on Computational Linguistics, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. Preprint, arXiv:2106.09685.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. arXiv preprint arXiv:2410.21276.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. Preprint, arXiv:2403.07974.

Hao Jiang, Nan Wei, Xiao Pan, Yu Li, Xiaochen Liang, Yi Tay, Jilan Wei, Xiaohui Yan, Xiaozhi Wang, Juanzi Li, et al. 2024a. LongBench: A bilingual, multitask benchmark for long context understanding. Transactions on Machine Learning Research.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Llmlingua: Compressing prompts for accelerated inference of large language models. Preprint, arXiv:2310.05736.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024b. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. Preprint, arXiv:2310.06839.

Pengcheng Jiang, Jiacheng Lin, Lang Cao, Runchu Tian, SeongKu Kang, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. Deepretrieval: Hacking real search engines and retrievers with large language models via reinforcement learning. Preprint, arXiv:2503.00223.

Tian Lan, Xiangdong Su, Xu Liu, Ruirui Wang, Ke Chang, Jiang Li, and Guanglai Gao. 2025. Mcbe: A multi-task chinese bias evaluation benchmark for large language models. arXiv preprint arXiv:2507.02088.

Taewhoo Lee, Chanwoong Yoon, Kyochul Jang, Donghyeon Lee, Minju Song, Hyunjae Kim, and Jaewoo Kang. 2024. Ethic: Evaluating large language models on long-context tasks with high information coverage. arXiv preprint arXiv:2410.16848.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. 2023. Loogle: Can long-context language models understand long contexts? arXiv preprint arXiv:2311.04939.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437.

Wanlong Liu, Junying Chen, Ke Ji, Li Zhou, Wenyu Chen, and Benyou Wang. 2024b. Rag-instruct: Boosting llms with diverse retrieval-augmented instructions. arXiv preprint arXiv:2501.00353.

Xuyang Liu, Ziming Wang, Yuhang Han, Yingyao Wang, Jiale Yuan, Jun Song, Bo Zheng, Linfeng Zhang, Siteng Huang, and Honggang Chen. 2025a. Global compression commander: Plug-and-play inference acceleration for high-resolution large vision-language models. arXiv preprint arXiv:2501.05179.

Xuyang Liu, Zichen Wen, Shaobo Wang, Junjie Chen, Zhishan Tao, Yubo Wang, Xiangqi Jin, Chang Zou, Yiyu Wang, Chenfei Liao, et al. 2025b. Shifting ai efficiency from model-centric to data-centric compression. arXiv preprint arXiv:2505.19147.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A Rossi, Seunghyun Yoon, and Hinrich Schütze. 2025. Nolima: Long-context evaluation beyond literal matching. arXiv preprint arXiv:2502.05167.

Rahmatullah Musawi and Sheng Lu. 2025. Towards contamination resistant benchmarks. arXiv preprint arXiv:2505.08389.

Milad Nasr, Nicholas Carlini, Matthew Roberts, Tianyi Zhang, Katherine Song, Úlfar Erlingsson, Nicolas Papernot, Chris Lee, and Colin Raffel. 2023. A comprehensive study of memorization in large language models. arXiv preprint arXiv:2307.06244.

Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2023. Proving test set contamination in black-box language models. In The Twelfth International Conference on Learning Representations.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. Preprint, arXiv:2403.12968.

Zehan Qi, Rongwu Xu, Zhijiang Guo, Cunxiang Wang, Hao Zhang, and Wei Xu. 2024. Long^2 rag: Evaluating long-context & long-form retrieval-augmented generation with key point recall. arXiv preprint arXiv:2410.23000.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. Preprint, arXiv:2409.05591.

Kun Qian, Shunji Wan, Claudia Tang, Youzhi Wang, Xuanming Zhang, Maximillian Chen, and Zhou Yu. 2024. Varbench: Robust language model benchmarking through dynamic variable perturbation. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 16131–16161.

Yilun Qiu, Tianhao Shi, Xiaoyan Zhao, Fengbin Zhu, Yang Zhang, and Fuli Feng. 2025. Latent inter-user difference modeling for llm personalization. arXiv preprint arXiv:2507.20849.

Qwen. 2025. Qwen2.5 technical report. Preprint, arXiv:2412.15115.

Tanmay Rajore, Nishanth Chandran, Sunayana Sitaram, Divya Gupta, Rahul Sharma, Kashish Mittal, and Manohar Swaminathan. 2024. Truce: Private benchmarking to prevent contamination and improve comparative evaluation of llms. arXiv preprint arXiv:2403.00393.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023a. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 10776–10787, Singapore. Association for Computational Linguistics.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023b. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. arXiv preprint arXiv:2310.18018.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison Over Long Language Sequences. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. arXiv preprint arXiv:2310.16789.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. arXiv preprint arXiv:2401.01313, 6.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. Transactions of the Association for Computational Linguistics, 10:539–554.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave No Document Behind: Benchmarking Long-Context LLMs with Extended Multi-Doc QA. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. 2025a. Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. arXiv preprint arXiv:2502.18017.

Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. 2025b. Vrag-rl: Empower vision-perception-based rag for visually rich information understanding via iterative reasoning with reinforcement learning. arXiv preprint arXiv:2505.22019.

Yifei Wang, Feng Xiong, Yong Wang, Linjing Li, Xiangxiang Chu, and Daniel Dajun Zeng. 2025c. Position bias mitigates position bias: Mitigate position bias through inter-position knowledge distillation. arXiv preprint arXiv:2508.15709.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Benjamin Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Venkat Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2024. LiveBench: A Challenging, Contamination-Limited LLM Benchmark. In The Thirteenth International Conference on Learning Representations.

Xiaobao Wu, Liangming Pan, Yuxi Xie, Ruiwen Zhou, Shuai Zhao, Yubo Ma, Mingzhe Du, Rui Mao, Anh Tuan Luu, and William Yang Wang. 2024. Antileakbench: Preventing data contamination by automatically constructing benchmarks with updated real-world knowledge. arXiv preprint arXiv:2412.13670.

Cheng Xu, Shuhao Guan, Derek Greene, and M.-Tahar Kechadi. 2024a. Benchmark Data Contamination of Large Language Models: A Survey. Preprint, arXiv:2406.04244.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. 2024b. Benchmark data contamination of large language models: A survey. Preprint, arXiv:2406.04244.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. 2024c. Benchmarking benchmark leakage in large language models. Preprint, arXiv:2404.18824.

Zhe Xu, Jiasheng Ye, Xiaoran Liu, Xiangyang Liu, Tianxiang Sun, Zhigeng Liu, Qipeng Guo, Linlin Li, Qun Liu, Xuanjing Huang, et al. 2024d. Detectiveqa: Evaluating long-context reasoning on detective novels. arXiv preprint arXiv:2409.02465.

Xinwei Yang, Zhaofeng Liu, Chen Huang, Jiashuai Zhang, Tong Zhang, Yifan Zhang, and Wenqiang Lei. 2025. Elaboration: A comprehensive benchmark on human-llm competitive programming. arXiv preprint arXiv:2505.16667.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tao Yuan, Xuefei Ning, Dong Zhou, Zhijie Yang, Shiyao Li, Minghui Zhuang, Zheyue Tan, Zhuyu Yao, Dahua Lin, Boxun Li, et al. 2024. Lv-eval: A balanced long-context benchmark with 5 length levels up to 256k. arXiv preprint arXiv:2402.05136.

Huixuan Zhang, Yun Lin, and Xiaojun Wan. 2024. PaCoST: Paired Confidence Significance Testing for Benchmark Contamination Detection in Large Language Models. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 1794–1809.

Jusheng Zhang, Yijia Fan, Wenjun Lin, Ruiqi Chen, Haoyi Jiang, Wenhao Chai, Jian Wang, and Keze Wang. 2025a. Gam-agent: Game-theoretic and uncertainty-aware collaboration for complex visual reasoning. arXiv preprint arXiv:2505.23399.

Jusheng Zhang, Zimeng Huang, Yijia Fan, Ningyuan Liu, Mingyan Li, Zhuojie Yang, Jiawei Yao, Jian Wang, and Keze Wang. 2025b. KABB: Knowledge-aware bayesian bandits for dynamic expert coordination in multi-agent systems. In Forty-second International Conference on Machine Learning.

Dawei Zhu, Xiyu Wei, Guangxiang Zhao, Wenhao Wu, Haosheng Zou, Junfeng Ran, Xun Wang, Lin Sun, Xiangzheng Zhang, and Sujian Li. 2025. Chain-of-thought matters: improving long-context language models with reasoning path supervision. arXiv preprint arXiv:2502.20790.

A Finetune Settings

We fine-tune four models on both datasets using one NVIDIA A100-SXM4-80GB GPU for 10 epochs. The fine-tuning process leverages the Unsloth library (Daniel Han and team, 2023) and incorporates LoRA (Hu et al., 2021) with a rank of 32 and an alpha of 32. The initial learning rate is set at 2e-5. We adopt a per-device training batch size of 12 and set the gradient accumulation steps to 4 for LLaMA-3.1-8B and Qwen3-8B. Meanwhile, for Phi-4 and Qwen2.5-7B, we use a per-device training batch size of 8 and set the gradient accumulation steps to 6.

B Prompt Templates

We present the two core prompt templates for our defense pipeline. Table 8 shows the CoT prompt used to identify critical evidence segments, while Table 9 guides the counterfactual rewriting process to generate contradictory alternatives.

Answer the question based on the given passages. The following are the passages:
{Context}
Answer the question based on the given passages.
Question: {Question}
Please first provide your answer in the format of Answer:[Your answer]. Then provide your reasoning process step-by-step.(Only include explicit clues)
At the end of each reasoning step, include a new line that specifies the key information or reference content used in that step.
Please ensure that the [reference content] you include is the complete original sentence or consecutive sentences from the text. Please do not change the punctuation. Do not use ellipses inside the sentence.
Follow this format:
Answer: [Your answer]
Step-by-step Reasoning:
1. [Reasoning step 1]
[replaced by your reference content]
2. [Reasoning step 2]
[replaced by your reference content]

Table 8: Chain-of-Thought Reasoning Prompt

You are a creative contrarian. Given the question below, and the original answer, first propose a concise alternative answer—that is, a plausible but intentionally misleading answer.
Followed are some sentences supporting the original answer, please rewrite them. When rewriting each sentence, modify only the parts necessary to support the antifact answer. Parts unrelated to the answer must keep their original meaning. Be sure that the modified evidence sentences are sufficient to answer the original question. Output must be strictly in the specified JSON format, with no additional text.
{
"answer": "<your antifact answer here, just provide the answer phrase, no need for complete sentence>",
"revised": [
"<rewritten sentence 1>",
"<rewritten sentence 2>",
]
}
Question:
{Question}
Original answer:
{Original Answer}
Sentences to rewrite:
{Numbered Sentences}

Table 9: Counterfactual Evidence Rewriting Prompt